

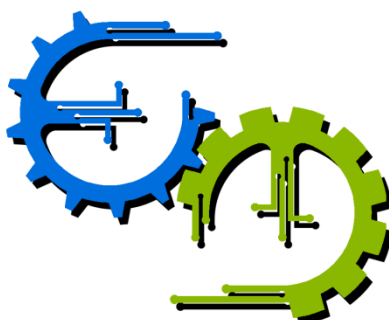


TRABALHO DE GRADUAÇÃO

**SISTEMA SEM FIO DE MONITORAMENTO DE
CONSUMO E QUALIDADE DE ENERGIA**

Por,
Claudio de Paula Donaté

Brasília, Fevereiro de 2016



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

TRABALHO DE GRADUAÇÃO

**SISTEMA SEM FIO DE MONITORAMENTO DE
CONSUMO E QUALIDADE DE ENERGIA**

POR,

Claudio de Paula Donaté

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro de Controle e Automação.

Banca Examinadora

Prof. Aida Alves Fadel, UnB/ENM (Orientadora) _____

Prof. Dianne Magalhães Viana, UnB/ENM _____

Brasília, Fevereiro de 2016

FICHA CATALOGRÁFICA

CLAUDIO, DE PAULA DONATÉ

SISTEMA SEM FIO DE MONITORAMENTO DE CONSUMO E QUALIDADE DE ENERGIA

[Distrito Federal] 2015.

xvii, 74p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2015). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.

1.Monitoramento

2.Qualidade

3.Sem fio

4.Energia

I. Mecatrônica/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

DONATÉ, CPD, (2015). Sistema Sem Fio de Monitoramento de Consumo e Qualidade de Energia. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº 023/2015, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 74p.

CESSÃO DE DIREITOS

AUTOR: Claudio de Paula Donaté.

SISTEMA SEM FIO DE MONITORAMENTO DE CONSUMO E QUALIDADE DE ENERGIA: Concepção, projeto e construção de um sistema para monitoramento do consumo e qualidade de energia.

GRAU: Engenheiro

ANO: 2015

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Claudio de Paula Donaté
SQN 316, Bloco K, Apto. 124 – Asa Norte.
70775-110 Brasília – DF – Brasil.

DEDICATÓRIA

Dedico este projeto à minha mãe. Sem ela, nada disso seria possível. Obrigado por estar sempre presente quando precisei.

Claudio de Paula Donaté

AGRADECIMENTOS

Agradeço a todos que me ajudaram a superar essa longa fase. Agradeço minha mãe, Neuza Maria de Paula e seu marido, Carlos Eduardo Barbosa Pimentel, por toda a ajuda dada ao texto. Minha orientadora Aida Fadel, obrigado pela ajuda, paciência e confiança. Meus agradecimentos também para Carlos Henrique da Silva Mendonça, que, muito gentilmente, cedeu seu laboratório para o processo de calibração, indispensável para a conclusão do projeto.

Obrigado por toda a ajuda. Acabou!

Claudio de Paula Donaté

RESUMO

O presente trabalho apresenta o processo de concepção, desenvolvimento e teste de um Sistema Sem Fio para Monitoramento de Consumo e Qualidade de Energia. O sistema proposto utiliza o CI CS5490 para medição de energia e faz uso de comunicação via protocolo *ZigBee* para a criação de uma rede de sensores sem fio com objetivo de permitir ao consumidor final uma avaliação de seu consumo e da qualidade da energia recebida, por meio de uma interface amigável também desenvolvida no escopo deste trabalho.

O protótipo desenvolvido apresentou resultados operacionais bastante satisfatórios.

Palavras Chave: monitoramento, qualidade, energia, sem fio.

ABSTRACT

This paper presents the process of designing, developing and testing a Wirelessly Monitoring Power Consumption and Quality. The proposed system uses the IC CS5490 for energy measurement and makes use of communication via the ZigBee protocol to create a wireless sensor network that will allow the user to evaluate his consumption levels, and the quality of energy he receives at his home through a user-friendly interface, also developed in the scope of this paper.

The prototype presented satisfactory operating results.

Keywords: monitoring; quality; wireless; energy;

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CENÁRIO PARA O DESENVOLVIMENTO DA PROPOSTA	1
1.2	OBJETIVOS	1
1.2.1	Objetivo Geral	1
1.2.2	Objetivos Específicos.....	2
1.3	JUSTIFICATIVA	2
1.4	ESTRUTURA DO TRABALHO	3
2	ESTUDO DAS TECNOLOGIAS EXISTENTES	4
2.1	FORMAS DE COMUNICAÇÃO SEM FIO	4
2.1.1	Bluetooth LE	4
2.1.2	Módulo nRF24L01+	5
2.1.3	Módulo Wi-Fi CC3000.....	6
2.1.4	Módulo XBee®.....	7
2.2	MICROCONTROLADOR	12
2.3	MICROCOMPUTADOR	12
2.3.1	BeagleBone	13
2.3.2	pcDuino Lite.....	13
3	PROPOSTA DE CONFIGURAÇÃO DO DISPOSITIVO	15
3.1	PARÂMETROS UTILIZADOS NA ESCOLHA DO PROTÓTIPO.....	15
3.2	DETALHAMENTO DO SISTEMA PROPOSTO	15
3.2.1	Esquemático com figura detalhando os componentes.....	15
3.3	CENTRAL DE PROCESSAMENTO	18
3.3.1	Microcomputador	18
3.3.2	Módulo De Comunicação Sem Fio	19
3.3.3	Relógio de tempo real.....	23
3.4	MÓDULO SENSOR	24

3.4.1	Unidade de processamento (MCU).....	25
3.4.2	Sensor de medição de consumo e qualidade da energia	26
3.4.3	Comunicação sem fio	29
4	APRESENTAÇÃO DO PROTÓTIPO	31
4.1	MÓDULOS SENSORES	31
4.2	CENTRAL DE CONTROLE	36
4.2.1	Script de controle do módulo XBee®.....	39
4.2.2	Servidor websocket.....	41
4.2.3	Script de acesso ao banco de dados	42
4.2.4	Banco de dados	42
4.2.5	Servidor HTTP	43
4.3	INTERFACE DE CONTROLE.....	44
5	COMISSIONAMENTO DO PROTÓTIPO	48
5.1	CALIBRAÇÃO CS5490	48
5.2	TESTES.....	52
6	CONCLUSÃO E RESULTADOS.....	55
6.1	VIABILIDADE E CUSTOS.....	56
7	REFERENCIAS BIBLIOGRAFICAS	59

LISTA DE FIGURAS

Figura 1 - Estrutura simplificada do sistema de medição proposto.	2
Figura 2 - Exemplo de dispositivo de integração Bluetooth (Mikrocontroller Praxis, s.d.)	5
Figura 3 - Comunicação nRF24L01+ (Fortytwoandnow, n.d.).....	6
Figura 4 - Módulo CC3000 (Sparkfun Eletronics, n.d.).....	7
Figura 5 - Camadas Zigbee (Home Toys, n.d.)	8
Figura 6 - Topologias de rede Zigbee (EE Times, n.d.).....	12
Figura 7 - BeagleBone (THE INTENTIONAL GEEK, n.d.).....	13
Figura 8 - pcDuino Lite (Sparkfun Electronics, n.d.).....	14
Figura 9 - Estrutura de funcionamento do Sistema de Medição	16
Figura 10 - Raspberry Pi® - modelo B+ (RASPBerry PI FOUNDATION, n.d.)	18
Figura 11 - Módulo XBee® serie 2 (Gravitech Us, n.d.).....	21
Figura 12 - XBee® API Packet (Million Bitz, n.d.).....	22
Figura 13 - RTC DS3231SN - Maxim Integrated (Home Coder, n.d.).....	24
Figura 14 - Circuito DS3231 básico (Rinky-Dink Electronics, n.d.).....	24
Figura 15 - Circuito mínimo para o CI atmega328P	26
Figura 16 - Circuito CS5490 (Cirrus Logic, 2013)	29
Figura 17 - Esquemático enfatizando os módulos sensores	31
Figura 18 - Placa de sensoriamento montada.....	34
Figura 19 - Placa de sensoriamento.....	34
Figura 20 - Módulo microcontrolador e de comunicação	35
Figura 21 - Circuito Sensor – Topo.....	35
Figura 22 - Circuito Sensor – Fundo	35
Figura 23 - Esquemático enfatizando a Central de processamento.....	36
Figura 24 - Circuito XBee®/RTC Central	37
Figura 25 - Esquemático XBee® / RTC Central.....	37
Figura 26 - Placa de circuito – Central de processamento	38
Figura 27 - Central de Processamento.....	38
Figura 28 - Esquemático enfatizando a página de controle	44
Figura 29 - Página de controle mostrando dados ao vivo	46
Figura 30 - Página de controle mostrando dados de histórico	46
Figura 31 - Página de controle exibindo os Nodes conectados.....	47

Figura 32 - <i>Dataflow</i> de Calibração (Cirrus Logic, 2012)	48
Figura 33 - Fluxograma completo de calibração (Cirrus Logic, 2012).....	49
Figura 34 - Fluxos de calibração AC Offset, DC Offset e No Load Offset (Cirrus Logic, 2012)	50
Figura 35 - Varivolt - Autotransformador monofásico	51
Figura 36 - Carga puramente resistiva	52
Figura 37 - Informações de consumo de recursos no Raspberry Pi®	54
Figura 38 - ESP8266MOD SMT (Shopclues, n.d.).....	56

LISTA DE TABELAS

Tabela 1 - Especificações Bluetooth LE	4
Tabela 2 - Especificações módulo nRF24L01+	5
Tabela 3 - Especificações Zigbee	8
Tabela 4 - Características BeagleBone.....	13
Tabela 5 - Características pcDuino Lite	13
Tabela 6 – Especificações do protótipo proposto.....	17
Tabela 7 - Especificações Raspberry Pi® B+	19
Tabela 8 - Especificações módulo XBee®.....	20
Tabela 9 - <i>Packet</i> enviado pela Central.....	22
Tabela 10 - Comparação entre modo API e AT	23
Tabela 11 - Especificações atmega328P	25
Tabela 12 - Estrutura da tabela <i>sensor_data</i>	42
Tabela 13 - Estrutura da tabela <i>sensor_history</i>	43
Tabela 14 - Potência nominal vs potência aferida.....	53
Tabela 15 - Preço dos componentes ativos	56

LISTA DE SÍMBOLOS

Símbolos Latinos

W	Watt	$[\text{kg.m}^2/\text{s}^3]$
V	Volt	$[\text{kg.m}^2/\text{A.s}^3]$
Hz	Hertz	$[1/\text{s}]$
J	Joule	$[\text{kg.m}^2/\text{s}^2]$
T	Temperatura	$[^\circ\text{C}]$
t	Tempo	$[\text{s}]$

Símbolos Gregos

Ω	Ohm	$[\text{kg.m}^2/\text{s}^3.\text{A}^2]$
----------	-----	---

Siglas

ACK	<i>Acknowledgement</i>
ADC	<i>Analog-Digital Converter</i>
API	<i>Application Programming Interface</i>
ARM	<i>Advanced RISC Machine</i>
BPSK	<i>Binary phase-shift keying</i>
CAN	<i>Controller Area Network</i>
CPU	<i>Central processing unit</i>
CI	<i>Circuito Integrado</i>
CMOS	<i>Complementary metal-oxide-semiconductor</i>
CSMA-CA	<i>Carrier sense multiple access with collision avoidance</i>
DDNS	<i>Dynamic DNS</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
DIO	<i>Digital Input/Output</i>
DSP	<i>Digital signal processor</i>
DSI	<i>Display Serial Interface</i>
DSSS	<i>Direct Sequence Spread Spectrum</i>
EEPROM	<i>Electrically-Erasable Programmable Read-Only Memory</i>
GPU	<i>Graphics processing unit</i>
GPIO	<i>General Purpose Input/Output</i>

GTS	<i>Guaranteed time slot</i>
HDMI	<i>High-Definition Multimedia Interface</i>
HTTP	<i>Hypertext Transfer Protocol</i>
I ² C	<i>Inter-Integrated Circuit</i>
I ² S	<i>Inter-IC Sound</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
JSON	<i>JavaScript Object Notation</i>
LSB	<i>Least significant bit</i>
MAC	<i>Media Access Control</i>
MCU	<i>Microcontrolador</i>
ND	<i>Node Discover</i>
NTP	<i>Network Time Protocol</i>
O-QPSK	<i>Offset quadrature phase-shift keying</i>
OSI	<i>Open Systems Interconnection</i>
PAN	<i>Personal Area Network</i>
PHY	<i>Physical Layer</i>
RAM	<i>Random-access memory</i>
RF	<i>Rádio Frequência</i>
RTC	<i>Real time clock</i>
RSSI	<i>Received signal strength indication</i>
SDRAM	<i>Synchronous dynamic random access memory</i>
SI	<i>Sistema Internacional</i>
SMD	<i>Semi Metallic Disc</i>
SMT	<i>Surface-mount technology</i>
SPI	<i>Serial Peripheral Interface</i>
SRAM	<i>Static Random Access Memory</i>
TC	<i>Transformador de corrente</i>
TCP	<i>Transmission Control Protocol</i>
TQFP	<i>Thin Quad Flat Package</i>
UART	<i>Universal asynchronous receiver/transmitter</i>
UDP	<i>User Datagram Protocol</i>
USB	<i>Universal Serial Bus</i>
WPAN	<i>Wireless Personal Area Networks</i>
ZDO	<i>ZigBee Device Object</i>

1 INTRODUÇÃO

1.1 Cenário para o desenvolvimento da proposta

Após diversos avanços na automação industrial e predial, o consumidor brasileiro está procurando cada vez mais serviços de automação residencial. Contudo, os sistemas disponíveis hoje, na sua maioria, são caros, de difícil aquisição e nem sempre facilmente operáveis. Quase sempre apresentam a necessidade de intervenção do usuário, utilizando poucos recursos para automatizar o processo.

Avanços recentes em tecnologia nos trouxeram dispositivos multifuncionais, como *notebooks*, *tablets*, *smartphones*, etc. Cada vez mais intrínseco no nosso dia a dia, esses dispositivos já são capazes de controlar grande parte dos aspectos das nossas vidas.

Por outro lado, a cada ano, o aumento da população mundial é acompanhado pelo crescimento do consumo de energia. Esse aumento, que hoje já mostra sinais de descontrole, vem acarretando diversos problemas, sendo o aquecimento global um dos mais discutidos, apontando para a intensa necessidade de nos voltarmos para uma forma de vida mais sustentável.

Foi pensando em atender a essa necessidade de sustentabilidade que esse projeto se iniciou, com o objetivo de proporcionar ao usuário o controle e monitoramento central do consumo de energia do domicílio em uma plataforma desenvolvida em ambiente web, em rede local, permitindo o acesso direto via navegador de internet.

Nesse cenário, o trabalho aqui exposto tem como objetivo o desenvolvimento de um sistema residencial de monitoramento e controle de energia. O sistema é proposto completamente sem fio, com a comunicação entre central e sensores sendo feita por rede Zigbee.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo geral do projeto é a proposição e desenvolvimento de um protótipo e interface de um Sistema Sem Fio de Monitoramento de Consumo e Qualidade de Energia baseado em características como simplicidade, custo reduzido e robustez, cuja estrutura de funcionamento pode ser entendida na Figura 1.

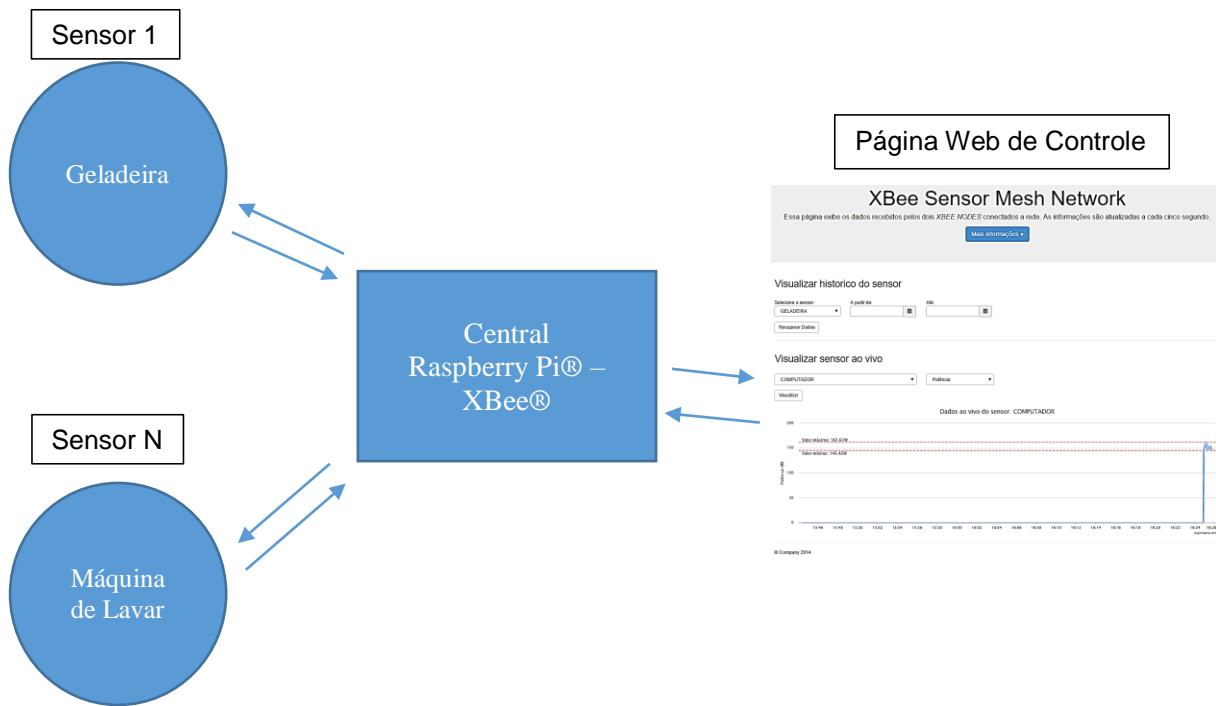


Figura 1 – Estrutura simplificada do sistema de medição proposto.

1.2.2 Objetivos Específicos

- Estudo das tecnologias existentes para determinação daquelas a serem utilizadas (sensoriamento, comunicação sem fio, interface amigável);
- Proposição de leiaute real para o sistema;
- Desenvolvimento do sistema / protótipo;
- Teste e avaliação do protótipo;
- Análise de custo.

1.3 Justificativa

A crise no setor de geração de energia agravada pela escassez dos recursos hídricos, que compõem a maior parte da matriz energética brasileira, impactada pelo regime de chuvas e o aumento do custo de geração causado pela complementação do atendimento da demanda com energia gerada em unidades termoeletricas, soma-se ao sentimento da sociedade brasileira de reconhecer a necessidade real de sistemas sustentáveis. Entretanto, na quase totalidade dos casos, o consumidor tem apenas como instrumento de controle o uso do interruptor e os dados fornecidos pela provedora de energia elétrica do local.

Isto vem mudando, com a consciência pública cada vez mais atenta aos desperdícios o que gera a necessidade de um sistema simples, barato e robusto de controle da energia

consumida. Atualmente, sistemas de monitoramento de energia não são incomuns, embora sejam ainda bastante desconhecidos pelo público, além de relativamente caros.

1.4 Estrutura do trabalho

O presente capítulo apresenta o cenário e justificativas para o desenvolvimento do trabalho, bem como seus objetivos.

O Capítulo 2 mostra o estudo feito das tecnologias disponíveis para o desenvolvimento do protótipo. Uma breve introdução é feita para as tecnologias consideradas, porém não escolhidas.

No Capítulo 3 os dispositivos e tecnologias usadas no protótipo são apresentados e detalhados.

O Capítulo 4 traz o detalhamento de cada módulo do sistema construído, assim como seu funcionamento interno e o pensamento por trás das escolhas feitas no desenvolvimento.

O Capítulo 5 descreve o processo de comissionamento do sistema, onde a calibração e os testes realizados são apresentados.

O Capítulo 6 apresenta as conclusões e resultados alcançados ao final do desenvolvimento do projeto, e uma análise de custo do produto final.

O Capítulo 7 contém a bibliografia do projeto.

2 ESTUDO DAS TECNOLOGIAS EXISTENTES

Neste capítulo serão apresentadas as tecnologias existentes para os subsistemas necessários ao desenvolvimento do dispositivo proposto, quais sejam: o sensoramento, a comunicação sem fio, microcontroladores e ferramentas para desenvolvimento da interface amigável.

2.1 Formas de comunicação sem fio

Várias formas de comunicação sem fio foram consideradas para o projeto com o protocolo *Zigbee* tendo sido escolhido por ser o mais próximo das necessidades do sistema. Abaixo, são expostas as soluções consideradas.

2.1.1 Bluetooth LE

O Bluetooth é uma tecnologia estabelecida há vários anos, sendo utilizada em milhares de dispositivos dos mais diversos tipos. Seu principal propósito é ser utilizado na criação de redes pessoais sem fio (*Wireless Personal Area Networks – WPANs*), com facilidade de conexão, pequeno raio de alcance (geralmente 10 metros) e média/alta velocidade de transferência.

Tabela 1 - Especificações Bluetooth LE

Especificação	Valor
Configuração de Rede	Ponto a multiponto
Alcance	10m em ambientes internos
Frequência de operação	2.4GHz
Velocidade	25Mbps
Tempo de conexão	~6s
Segurança	Existe a nível de protocolo

Com o tipo de conexão voltado principalmente para comunicação ponto a multiponto, somente há pouco tempo dando os primeiros passos na criação de redes em topologia *mesh*, o alcance limitado e não existência de topologias *mesh* já bem estabelecidas, foram as principais razões para que esse protocolo fosse deixado de lado no projeto, pois todas as outras características eram suficientes. Porém, sem a possibilidade de redirecionamento de mensagens, o sistema não seria concretizado. Conforme mostrado na Figura 2.



Figura 2 - Exemplo de dispositivo de integração Bluetooth (Mikrocontroller Praxis, s.d.)

2.1.2 Módulo nRF24L01+

O módulo RF nRF24L01+ é um transmissor baseado no chip da Nordic Semiconductor desenvolvido para aplicações sem fio (*wireless*) que utiliza o protocolo *Enhanced ShockBurst™* na sua camada de enlace de dados. Ele é configurado através da interface SPI (*Serial Peripheral Interface* - Interface Periférica Serial) e contém até 128 canais que operam na faixa 2.4GHz.

Tabela 2 - Especificações módulo nRF24L01+

Especificação	Valor
Tensão	1.9 a 3.6V (3.3V recomendado)
Taxas de transmissão	256Kbps, 1Mbps ou 2Mbps
Frequência de operação	2.4GHz
Multicanais de operação	128
Corrente durante a transmissão	11.3mA
Corrente durante a recepção	12.3mA
Corrente em repouso	900nA
Potência de transmissão programável em	0 (máxima), -6, -12 ou -18dBm
Sensibilidade de recepção	-82dBm a 2Mbps
Buffer	1 até 32 bytes de dados por vez
Temperatura de trabalho	-40 a 85°C
Alcance de transmissão com antena impressa	10m em ambientes internos

Esse módulo tem uma capacidade de receber dados de até seis transmissores simultaneamente, sem causar interferência. Cada Módulo RF nRF24L01+ contém um

conjunto de seis pipelines (canais de comunicação) de dados paralelos com endereços únicos. Um "Tubo" de dados é um canal lógico que possui um endereço físico único no canal de RF decodificado no chip nRF24L01+. Por essa razão, para estabelecer uma comunicação entre o transmissor (Tx) e receptor (Rx), é preciso configurar o mesmo endereço em ambos os módulos.

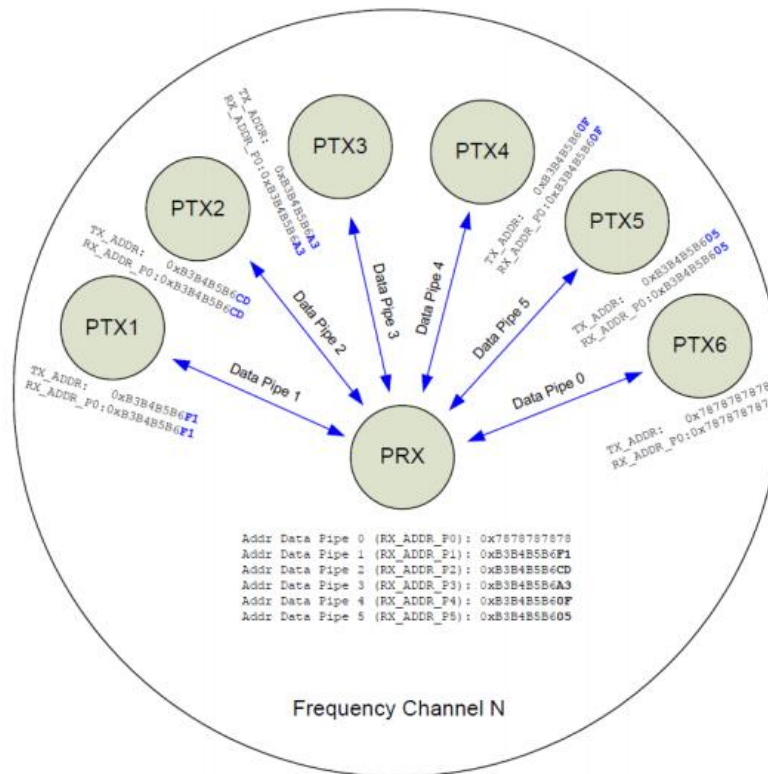


Figura 3 - Comunicação nRF24L01+ (Fortytwoandnow, n.d.)

Sem a possibilidade de redirecionamento de dados, criação de redes em topologia *mesh* e apenas 6 dispositivos conectados a uma central, o módulo foi rapidamente descartado.

2.1.3 Módulo Wi-Fi CC3000

Wi-Fi, com certeza, é a rede sem fio mais utilizada no mundo atual. É quase sempre possível encontrar diversas redes Wi-Fi em qualquer área urbana, desde shopping centers, cafés, e, principalmente, dentro das residências. Essa infraestrutura já existente cria um grande potencial para produtos embarcados sem fio, focados em automação, sensoriamento e entretenimento.

Visando este mercado, com foco principal em sistemas embarcados, a Texas Instruments criou o CC3000, um módulo de relativo baixo custo (~US\$12.50) quando se pensa em Wi-Fi, que implementa toda a parte complexa do protocolo de comunicação IEEE 802.11 e ainda complementa implementando as camadas de rede e de transporte. No nível físico da rede, o módulo necessita apenas de uma antena externa.

O CC3000 tem suporte para IPv4 com DHCP e suporta até quatro sockets simultâneos (UDP e TCP). Por ser focado em aplicações embarcadas, consome menos de 5uA no modo *Shut Down*. Além disso, pode ser alimentado diretamente por uma bateria e possui I/Os com alimentação separada, simplificando a interface com dispositivos que trabalham com tensão diferente do CC3000. A comunicação com o microcontrolador/microprocessador é realizada através de uma interface SPI (até 16MHz), periférico comumente presente até nos mais modestos microcontroladores. O CC3000 tem uma taxa máxima de transmissão de dados de 4Mbps para sockets TCP e 7Mbps para sockets UDP.



Figura 4 - Módulo CC3000 (Sparkfun Eletronics, n.d.)

Com a característica do CC3000 de funcionar apenas em modo infraestrutura, um ponto de acesso precisaria ser instalado sempre que a distância entre node e roteador fosse maior que o alcance do módulo. Por essa razão, a utilização desse módulo foi descartada.

2.1.4 Módulo XBee®

2.1.4.1 O Padrão ZigBee

O ZigBee é um padrão desenvolvido pela ZigBee Alliance, uma associação formada por grandes empresas do ramo de tecnologia, que trabalham em conjunto no desenvolvimento dessa alternativa de comunicação sem fio. Conforme a própria ZigBee Alliance (2008) essa tecnologia é fruto de esforços de empresas com necessidades em comum em torno de um padrão de comunicação simples e robusto.

De acordo com (Farahani, 2008), o principal alvo do padrão ZigBee são aplicações não muito complexas que necessitam de baixo consumo de energia, baixas taxas de transferências e baixo custo.

O ZigBee utiliza a norma IEEE 802.15.4, homologada em maio de 2003, como camada física e camada de acesso ao meio, onde é proposta uma comunicação de dispositivos sem fio com baixa taxa de transferência em uma PAN (*Personal Area Network*).

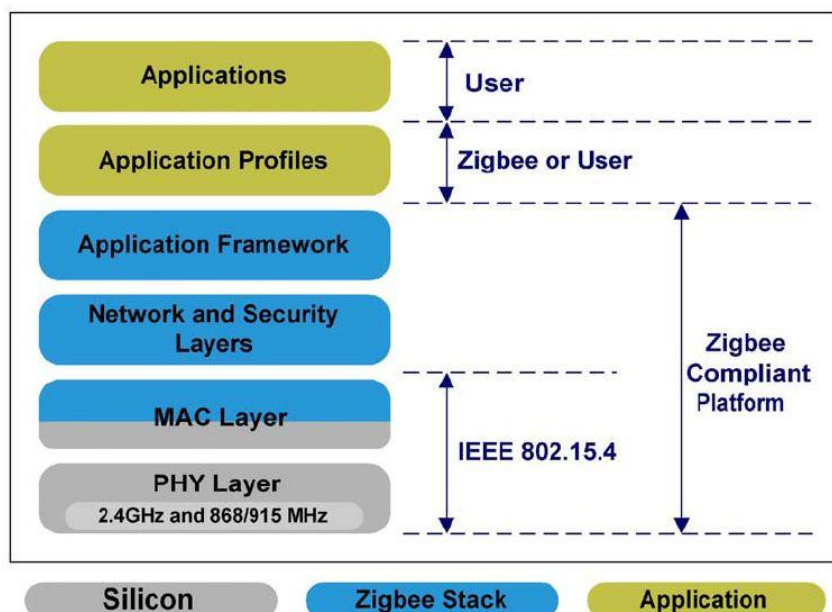


Figura 5 - Camadas Zigbee (Home Toys, n.d.)

2.1.4.2 Camada Física

Segundo o modelo OSI, a camada física, também conhecida como PHY (*Physical Layer*), tem a responsabilidade de tratar a transmissão dos dados brutos por meio de um canal de comunicação. Para isso, é preciso definir uma série de parâmetros, como, por exemplo, modulação e taxa de transferência, necessários para estabelecer esse vínculo.

O padrão IEEE 802.15.4 oferece diferentes faixas de frequência, que são aplicadas de acordo com a região. Nos Estados Unidos, a frequência utilizada é de 902–928MHz, possibilitando uma taxa de 40kbps. Na Europa, usa-se a frequência 868–868.6MHz, que atinge uma taxa de 20kbps. Já a frequência 2400–2483.5MHz é usada em todo o mundo, a taxa nesse caso é de 250kbps. A modulação também varia de acordo com a região, conforme a tabela abaixo.

Tabela 3 - Especificações Zigbee

Camada Física	Frequência	Canais	Velocidade (Kbps)	Modulação
868/915 MHz	868-870MHz	0	20	BPSK
	902-928MHz	1-10	40	BPSK

2.4 GHz	2.4-2.4835GHz	11-26	250	O-QPSK
---------	---------------	-------	-----	--------

A camada de controle de acesso ao meio (MAC) tem as seguintes responsabilidades: gerenciar o acesso ao canal, sincronização de *beacons* (baliza), validação/reconhecimento do quadro e de associação/desassociação de dispositivos na rede.

Segundo (Farahani, 2008), existem dois métodos de acesso ao canal: um baseado em contenção e outro livre de contenção.

No modelo baseado em contenção, o dispositivo começa a transmitir imediatamente, se se verificar que o meio está livre, utilizando o mecanismo CSMA-CA, que possui, em alguns casos, um tempo de contenção para iniciar a transmissão efetiva dos dados.

No método livre de contenção, o coordenador da PAN define um intervalo de tempo para cada dispositivo. Esse intervalo, chamado de *guaranteed time slot* (GTS), possibilita ao dispositivo realizar a transmissão sem a necessidade do protocolo CSMA-CA, logo, sem necessidade de contenção. Porém, é necessário que todos os dispositivos da rede estejam sincronizados para que esse método funcione. Isto é feito através de um pacote especial de controle chamado de *beacon*, que é enviado pelo coordenador.

Para o ZigBee, a grande vantagem do modelo livre de contenção, também conhecido como *beaconned*, é a economia de energia, pois o dispositivo pode ficar no estado dormente (*sleep*) enquanto não estiver no seu determinado slot. Já o modo que utiliza CSMA-CA, ou modo *non-beaconned*, é necessário que os dispositivos do tipo roteador estejam sempre ativos, impossibilitando poupar energia com a funcionalidade *sleep*.

2.1.4.3 Camada de Rede

A camada de rede é responsável por duas funções: o transporte de dados e também o suporte às aplicações dos dispositivos *ZigBee*. Segundo (Teixeira, 2006), algumas funcionalidades foram-lhe concedidas para desempenhar esse papel:

- *Network Scan*: Capacidade de um dispositivo de detectar um ou mais canais ativos em sua faixa de comunicação;
- *Creating/Joining* a PAN: Criar uma rede local e ingressar em uma já existente;
- *Device Discovery*: Capacidade de encontrar dispositivos sobre o canal ativo na PAN;
- *Service Discovery*: Descoberta de um serviço e a capacidade de determinar quais os serviços são suportados pelos dispositivos dentro de uma rede;

- *Binding*: Capacidade de se comunicar no nível da aplicação com outro dispositivo da rede.

2.1.4.4 Camada de Aplicação

A camada de aplicação é a de mais alto nível definida pela especificação ZigBee. Ela pode ser dividida em três elementos: subcamada de suporte à aplicação (APS), *Application Framework* e *ZigBee Device Object*.

Subcamada de Suporte à Aplicação (APS): fornece uma interface entre a camada de rede e a aplicação. É responsável por gerenciar a tabela de ligação, mantendo uma base de dados com os dispositivos conectados na rede. Também é sua função prover a transmissão de dados (PDUs) entre dois dispositivos na mesma rede. Para isso ele realiza a fragmentação e remontagem das PDUs, além dos serviços de segurança.

Application Framework: é o ambiente onde são alocados os objetos de aplicação ZigBee (*Application Objects*). Esses objetos possuem funções definidas pelos fabricantes, incluindo primitivas de serviço de dados como *request*, *response*, *confirm*. Cada objeto representa um tipo (ou profile) de aplicação diferente, que é definido para um único dispositivo ZigBee. É possível armazenar até 240 objetos de aplicação distintos em um dispositivo.

ZigBee Device Object (ZDO): implementa o profile do dispositivo proporcionando as funcionalidades básicas que permitem a comunicação entre a APS e os objetos de aplicação. Também é responsável por descobrir e identificar os serviços oferecidos por novos dispositivos na rede.

Application Profiles são definições de formatos de mensagens e processamento de ações que possibilitam a interoperabilidade entre dispositivos. É obrigatório que qualquer dispositivo na rede ZigBee implemente um profile.

Já os *Application Objects* encapsulam um conjunto de atributos que representam o estado do dispositivo e também funcionalidades (serviços) para ler/escrever nesses atributos.

2.1.4.5 Características

O padrão ZigBee possui diversas funcionalidades que foram desenvolvidas para atender às necessidades do mercado, principalmente da área de automação. Algumas das principais características do ZigBee são:

- Baixo consumo de energia: possui a habilidade de operar no modo *sleep*, podendo reduzir o consumo de energia de 70mA para 6uA;
- Baixo custo: possui uma pilha de protocolo de fácil implementação;
- Grande quantidade de nós: é possível ter 65535 nós em uma única rede;

- Diferentes topologias de rede: estrela, árvore ou *mesh*;
- Baixa latência: tempo pequeno de ligação à rede e rapidez na transição do modo de espera (*sleep*) para o modo ativo (30ms ou menos);
- Dois modos de operação da rede: beaconing (2 ns ou 1 n) e non-beaconing;
- Segurança e confiabilidade: possui recursos de encriptação com a implementação do padrão AES (*Advanced Encryption Standard*) de 128 bits;

2.1.4.6 Funcionamento

Para entender o funcionamento de uma rede ZigBee, faz-se necessário conhecer os três tipos de nós que a rede pode possuir. Os três tipos de nó são: roteador, dispositivo final ou coordenador.

O dispositivo coordenador (*coordinator*) é o agente central da rede, o nó que cria e gerencia uma rede ZigBee. Existe apenas um único coordenador por rede, sendo ele o responsável por concentrar as informações de interesse da aplicação. Por isso, ele possui uma maior capacidade computacional. É o coordenador que determina o identificador da rede (PAN ID) e este valor deverá ser utilizado por todos os equipamentos que desejam fazer parte da mesma rede.

Os dispositivos roteadores (*router*) possuem o importante papel de redirecionar os pacotes entre os nós da rede que não conseguem se comunicar diretamente. Isso permite a expansão da rede do ponto de vista físico.

Dispositivos finais (*end devices*) são aqueles que desempenham as funções de sensores ou atuadores da rede. Esses dispositivos podem se comunicar somente com os roteadores e o coordenador.

Uma das principais funcionalidades dos dispositivos finais é a possibilidade de operar no modo *sleep*, o que reduz de forma significativa o consumo de energia elétrica. Esse tipo de dispositivo é projetado para ficar a maior parte do tempo nesse modo, sendo que a transição para o modo ativo é extremamente rápida, cerca de 30 milissegundos.

2.1.4.7 Topologias de rede

De acordo com (Farahani, 2008), as aplicações que utilizam como base o IEEE 802.15.4 podem suportar as topologias estrela e ponto-a-ponto, sendo que nesta última é possível formar redes mais complexas como árvores e malha.

Na topologia estrela (*star*) diversos dispositivos se comunicam apenas com o coordenador da rede. Nesta topologia os dispositivos finais não conseguem se comunicar entre si, somente

por intermédio do coordenador, por esse motivo ela é utilizada geralmente em aplicações de baixa complexidade.

Na topologia do tipo ponto-a-ponto, os dispositivos de uma mesma rede podem se comunicar entre si diretamente, sem a necessidade do coordenador. Nesse caso, é possível organizar a rede em duas outras topologias: árvore ou malha.

Nas redes em árvore (*cluster tree*) é utilizada uma estratégia de roteamento hierárquico. Nodes finais podem se conectar apenas com o coordenador e roteadores e são chamados de filhos. Apenas o coordenador e roteadores podem ter filhos. Dispositivos finais (filhos) podem se comunicar apenas com seu pai (coordenador ou roteador). Com a utilização dos roteadores é possível expandir a rede geograficamente.

A rede em malha (*mesh*) é considerada uma extensão da topologia em árvore. Nela os roteadores podem se comunicar diretamente sem a necessidade de passar pelo coordenador. Com isso é possível expandir a rede indefinidamente, apenas acrescentando mais roteadores. Outra vantagem é a redundância na comunicação: caso um roteador perca a conexão, é possível redirecionar as mensagens por uma rota alternativa.

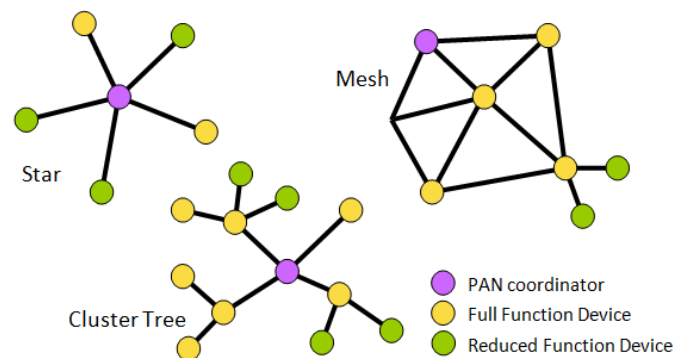


Figura 6 – Topologias de rede Zigbee (EE Times, n.d.)

2.2 Microcontrolador

Existem no mercado diversas opções de microcontroladores. O estudo prévio feito nessa área foi limitado, visto que o aluno já possuía conhecimento na plataforma atmega/Arduino, tendo plena ciência da capacidade desse CI. Nenhuma alternativa foi seriamente considerada.

2.3 Microcomputador

Para o microcomputador também existem diversas opções, sendo que o Raspberry Pi® foi escolhido, baseado principalmente no baixo custo e enorme comunidade de desenvolvedores.

2.3.1 BeagleBone

O BeagleBone é um microcomputador baseado no processador ARM Cortex-A8, que trabalha a 720MHz. Ele, como o Raspberry Pi®, roda uma distribuição Linux própria.

Tabela 4 - Características BeagleBone

Especificação	Valor
Processador	AM335x 720MHz ARM® Cortex-A8
Memória Flash	2GB 8-bit eMMC
Processadores secundários	Acelerador de gráficos 3D, acelerador de ponto flutuante NEON, 2x microcontrolador PRU de 32-bit
Conectividade	USB, Ethernet, micro HDMI
GPIO	2x I2C; 5x UART; I2S; SPI; CAN; 6x 3.3V GPIO; 7x ADC

Os dois grandes pontos que impediram a utilização do BeagleBone foi seu alto preço, US\$ 54,00 e por ser relativamente novo no mercado.

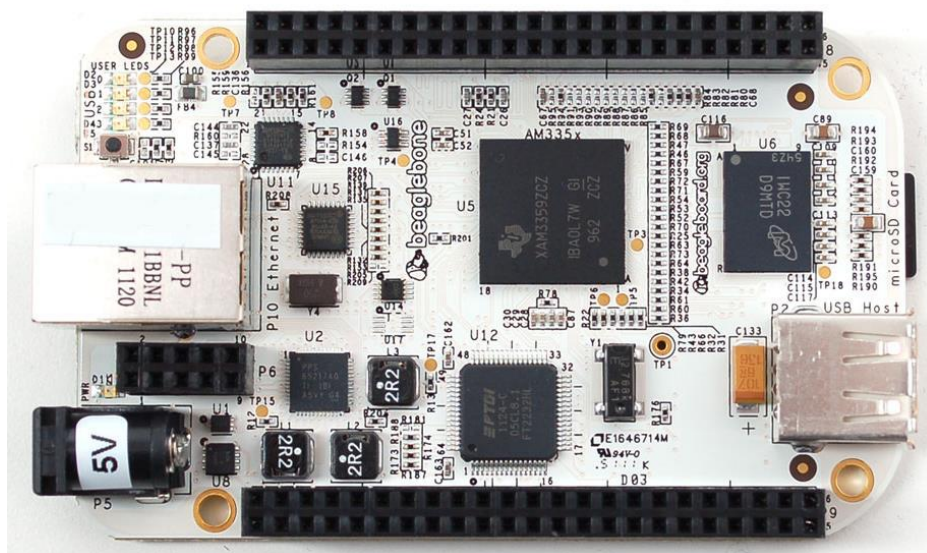


Figura 7 - BeagleBone (THE INTENTIONAL GEEK, n.d.)

2.3.2 pcDuino Lite

O pcDuino é um microcomputador baseado no processador ARM Cortex-A8, que trabalha a 1GHz. Ele, diferentemente do Raspberry Pi®, roda uma distribuição Linux aberta, mais especificadamente o Ubuntu 12.04.

Tabela 5 - Características pcDuino Lite

Especificação	Valor
Processador	ARM® Cortex-A8 @ 1GHz
Memória RAM	512 MB
GPU	OpenGL ES2.0, OpenVG 1.1 Mali 400 core
Conectividade	USB, Ethernet, HDMI
GPIO	I2C, UART, I2S, SPI, CAN, GPIO, ADC

Os dois grandes pontos que impediram a utilização do pcDuino Lite foi sua pequena comunidade de desenvolvedores e a pouca disponibilidade para compra.

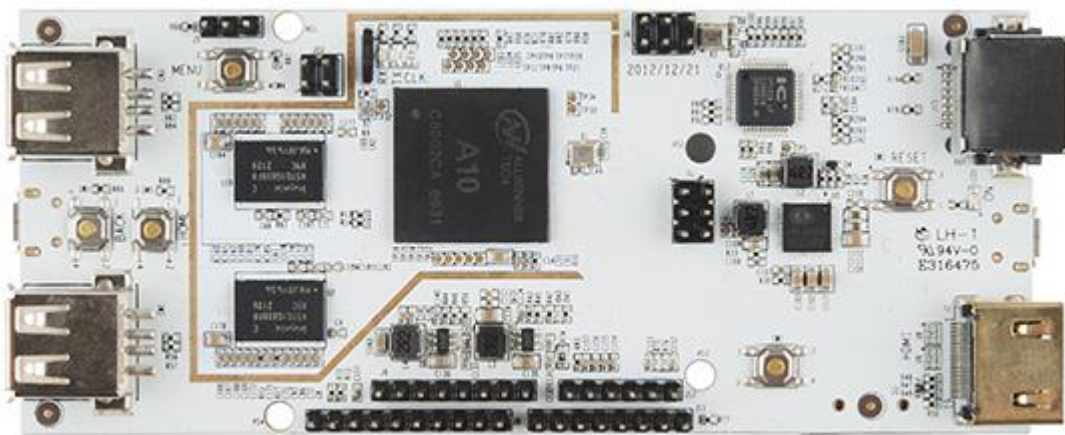


Figura 8 - pcDuino Lite (Sparkfun Electronics, n.d.)

3 PROPOSTA DE CONFIGURAÇÃO DO DISPOSITIVO

3.1 Parâmetros utilizados na escolha do protótipo

O parâmetro principal utilizado na escolha dos dispositivos foi a independência de conexão com a internet. Decidiu-se que o sistema deveria continuar funcionando normalmente mesmo que não existisse conexão com a internet. Além disso, o sistema deveria automaticamente rotear as mensagens de qualquer módulo sensor que estivesse fora do alcance direto da Central. Levando em consideração esses dois pontos, a solução encontrada foi a utilização do protocolo ZigBee, sendo o mesmo implementado com o uso das placas XBee®.

A utilização das placas XBee®, que apesar de permitirem que o sistema seja totalmente autossuficiente, elevou o custo final do projeto, visto que o módulo em si custa US\$ 17,50 na sua versão SMT¹.

3.2 Detalhamento do sistema proposto

3.2.1 Esquemático com figura detalhando os componentes

Podemos subdividir o sistema em dois módulos distintos, o de controle e o de sensoriamento. O módulo de controle, a partir de agora referido como “Central”, é onde todo o controle do sistema é feito. O módulo sensor, a partir de agora será referido como “Node” e como o nome já diz, é onde o sensoriamento acontece.

O esquemático completo do sistema proposto é mostrado na Figura 9, onde podem ser vistos cada Node conectado na rede, a Central de processamento, a página de controle e todas as mensagens trocadas no sistema.

As especificações de todos os componentes de cada subsistema estão listadas na Tabela 6, com suas funções e características detalhadas no texto anterior.

¹ <http://www.digi.com/products/models/xb24cz7pis-004>

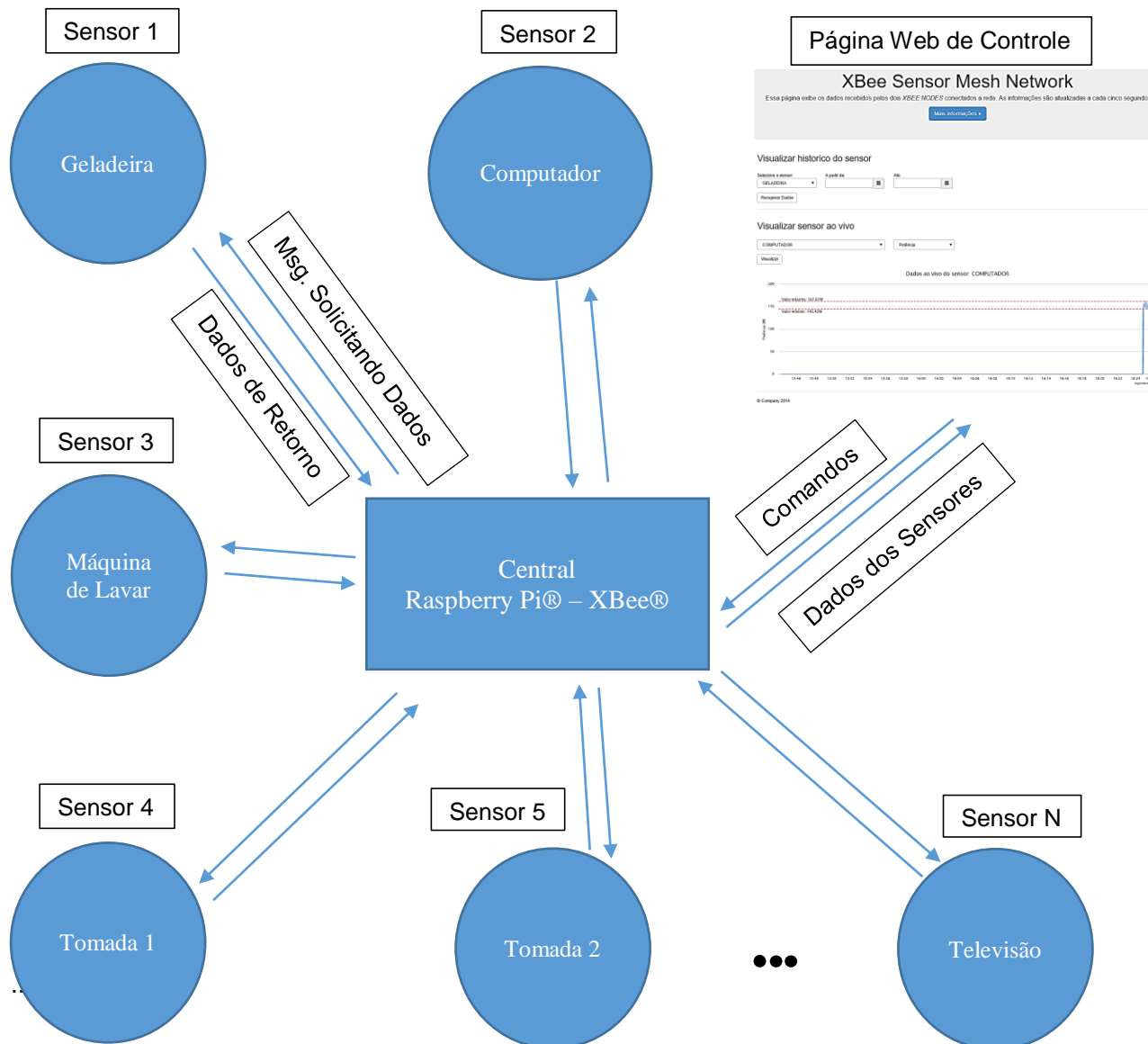


Figura 9 – Estrutura de funcionamento do Sistema de Medição

- Módulo sensor
 - Faz as medições de potência, tensão, corrente, fator de potência e temperatura de cada aparelho conectado;
 - Controla a comunicação com a Central, respondendo a solicitações.
- Central de processamento
 - Solicita e processa dados vindos dos módulos sensores;
 - Gerencia rede sem fio;
 - Gerencia histórico de dados;

- Gerencia página web de controle.
- Mensagem solicitando dados
 - Se o número de sensores é menos que cinco: Central envia mensagem a cada cinco segundos solicitando dados de medição dos sensores;
 - Se o número de sensores é maior que cinco: Central envia mensagem a cada dez segundos solicitando dados de medição dos sensores.
- Dados de retorno
 - O módulo sensor envia o cálculo da potência, tensão, corrente, fator de potência e temperatura para a central.
- Comandos
 - A página web de controle solicita inicialmente informações a Central referente aos módulos sensores conectados e sobre o histórico de todos módulos sensores já utilizados;
 - O usuário pode solicitar dados ao vivo de qualquer módulo sensor conectado ou sobre o histórico de medições;
 - O usuário pode alterar o nome de qualquer módulo sensor conectado ou desativar a gravação de dados de um determinado módulo sensor.
- Dados dos sensores
 - A Central envia a página web de controle os dados do módulo sensor selecionado.

Tabela 6 – Especificações do protótipo proposto

Sistema de Monitoramento sem Fio de Consumo e Qualidade de Energia	
Comunicação	Sem fio - ZigBee
Interface	Página web de controle
Tensão de entrada	De 90VAC até 250VAC
Corrente máxima	15A
Potência ativa máxima	3750W
Consumo elétrico Máximo	Central: 3W, Módulo Sensor: 0,5W
Consumo elétrico Médio	Central: 1,5W, Módulo Sensor: 0,3W
Memória	8GB – Aprox. 1 ano de histórico de dados

Processador	ARMv6 BCM2835 @ 800MHz
Memória RAM	512MB
Num. Máximo de módulos sensores	35 módulos sensores

3.3 Central de processamento

A Central é composta por um microcomputador, um módulo de comunicação sem fio e um *real time clock* (RTC²)

3.3.1 Microcomputador

O microcomputador escolhido para a Central foi o Raspberry Pi® modelo B+. A escolha foi feita principalmente pelo baixo custo dessa solução, acompanhado de um poder de processamento adequado para o propósito e baixo consumo de energia. O modelo B+, em comparação com o modelo A, possui porta de comunicação ethernet, indispensável para o sistema.

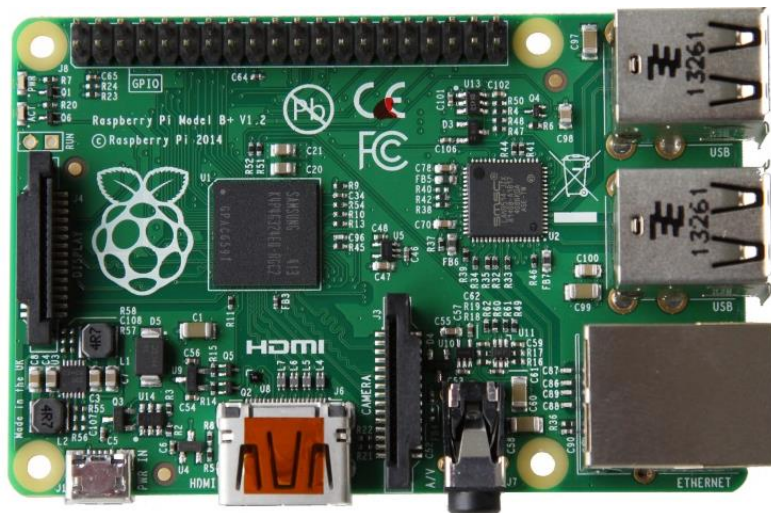


Figura 10 - Raspberry Pi® - modelo B+ (RASPBERRY PI FOUNDATION, n.d.)

O Raspberry Pi® modelo B+, possui:

- Entrada para cartão microSD
- Saída de vídeo HDMI
- Quatro portas USB
- 40 pinos de expansão para GPIO, I2C, UART, etc

² http://pt.wikipedia.org/wiki/Relógio_de_tempo_real

- Conector de áudio de 3.5mm
- Porta de interface para câmera (CSI-2)
- Porta de interface para LCD (DSI)
- Uma entrada microUSB para energia
- Uma porta ethernet

Tabela 7 - Especificações Raspberry Pi® B+

Especificação	Valor
SoC	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM, e USB)
CPU	ARMv6 BCM2835 @ 700MHz
Memória (SDRAM)	512 MB (compartilhada com a GPU)
Periféricos	27x GPIO, UART, I ² C bus, SPI bus, I ² S audio, +3.3 V, +5 V, terra
Consumo Máximo	600 mA (3W)
Alimentação	5V/2A via microUSB
Tamanho	85 mm × 56 mm x 17 mm
Peso	45 g

Uma grande vantagem do Raspberry Pi® é que o mesmo utiliza como sistema operacional uma distribuição Linux própria, podendo assim, facilmente ser programado de diversas formas utilizando as mais variadas linguagens de programação.

Outro ponto positivo é a grande comunidade reunida em torno do Raspberry Pi®, diferente de qualquer outro sistema equivalente. Isso garante que virtualmente qualquer erro encontrado terá solução ou alternativa viável.

3.3.2 Módulo De Comunicação Sem Fio

A ideia inicial do projeto sempre foi a utilização de comunicação por protocolo Zigbee, que opera na especificação IEEE 802.15.4.

O modulo escolhido para tal função foi o XBee® versão 2, produzido pela *Digi International*³.

Existem diversas possíveis escolhas para o modulo de comunicação Zigbee, porém, a grande maioria se limita a fornecer o chip de circuito integrado e referências bibliográficas sobre o produto. Muitas vezes com uma API (*Application Programming Interface*) confusa e pouco explanada, e, além disso, com ferramentas de desenvolvimento pagas e kits de

³ <http://www.digi.com/>

desenvolvimentos extremamente caros. É nesse ponto que os módulos XBee® têm sua grande vantagem, os mesmos são oferecidos como soluções fechadas, como podemos ver na Figura 11. O módulo já é aprovado no Brasil pela ANATEL⁴ e possui diversas certificações de emissões eletromagnéticas, não sendo necessária a homologação própria do produto final.

Tabela 8 - Especificações módulo XBee®

Especificação	Valor
Plataforma	XBee® ZB
Velocidade de comunicação	250Kbps
Alcance interno	40m
Alcance externo / Linha de visão	120m
Potência de transferência	1.25mW (+1dBm) / 2mW (+3dBm) modo boost
Interface I/O	3.3V CMOS UART, ADC, DIO
Modos de configuração	Comandos API ou AT, local ou over-the-air
Frequência de banda	2.4GHz
Imunidade a interferência	DSSS (Direct Sequence Spread Spectrum)
Velocidade de transferência serial	1200bps - 1Mbps
Entradas ADC	10-bit ADC
I/O Digitais	10
Opções de antena	Chip, Wire Whip, U.FL, RPSMA
Temperatura de operação	-40°C até 85°C, 0-95% de humidade sem condensação
Criptografia	128-bit AES
Tensão de alimentação	2.1 - 3.6VDC
Corrente de transmissão	40mA (@3.3 V)
Corrente de recebimento	38mA / 40mA em modo boost @ 3.3VDC
Corrente de Power-Down	<1uA @ 25° C

O tempo de desenvolvimento utilizando o módulo XBee® foi curto, em comparação com as outras soluções disponíveis. Pela utilização de pacotes de comunicação, enviados e recebidos pela porta serial de qualquer micro controlador, o uso se limita a montar e enviar os de saída e interpretar os de chegada. Toda lógica interna, comunicação entre as camadas de

⁴ <http://sistemas.anatel.gov.br/sgch/Certificado/HomologacaoPICC.asp?NumRFGCT=80410>

aplicação, envio de ACK⁵, reenvio de mensagens não recebidas, entre outros, é feito de forma transparente.

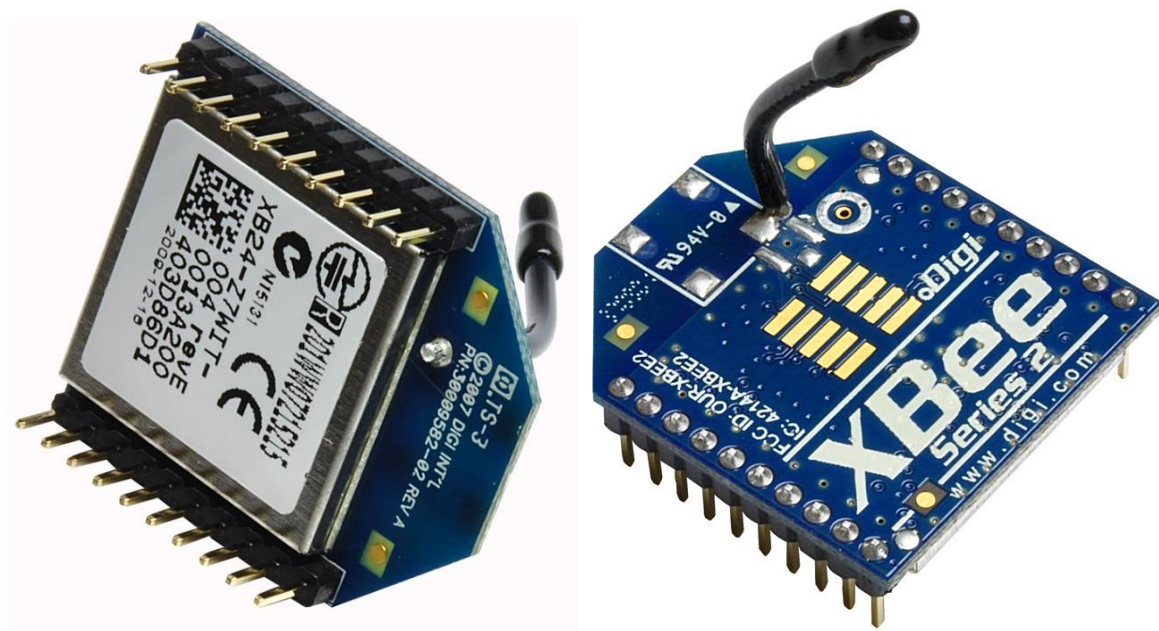


Figura 11 – Módulo XBee® serie 2 (Gravitech Us, n.d.)

A utilização do protocolo Zigbee foi escolhida principalmente pela possibilidade de criação de redes em topologia *MESH*, com a capacidade de redirecionamento das mensagens. Caso um Node não consiga comunicação direta com a “Central”, a mensagem é redirecionada automaticamente por “Nodes” que estejam entre a origem e o destino. Esta funcionalidade deixa a rede extremamente robusta e flexível, não sendo necessário que os Nodes fiquem todos ao alcance direto da Central.

Como já foi dito na seção 2.1.4.6, os módulos XBee® trabalham em três configurações diferentes: *End-device*, Roteador e Coordenador. Coordenadores são dispositivos responsáveis pela inicialização, distribuição de endereços, manutenção da Rede, reconhecimento de todos os nós, entre outras funções, podendo servir como ponte entre várias outras Redes ZigBee. Somente um é permitido em cada rede. O roteador tem as características de um Nó normal na Rede, mas com poderes extras de também exercer a função de roteador intermediário entre Nós, sem precisar do Coordenador. Por intermédio de um roteador, uma Rede ZigBee pode ser expandida, e assim ter mais alcance. Na prática um roteador pode ser usado para amplificar o sinal da Rede entre andares de uma casa. Já o *end-device* é onde os atuadores ou sensores são normalmente colocados. Ele é o nó que consome menos energia, pois pode ficar a maior parte do tempo em modo *sleep*.

⁵ [http://en.wikipedia.org/wiki/Acknowledgement_\(data_networks\)](http://en.wikipedia.org/wiki/Acknowledgement_(data_networks))

Um *packet* tem o formato e opções mostrados na Figura 11. Na Tabela 9 é mostrado o *packet* que a Central envia para cada Node solicitando um novo dado.

Tabela 9 - *Packet* enviado pela Central

7E	00	0C	00	01	00	13	A2	00	40	86	98	12	00	01	D8
Byte Início	Tamanho		Tipo Frame	ID do Frame	Endereço de 64 bits								Dado Enviado		Checksum

Frame Fields		Offset	Example	Description
A P I P a c k e t	Start Delimiter	0	0x7E	
	Length	MSB 1	0x00	Number of bytes between the length and the checksum
		LSB 2	0x16	
	Frame-specific Data			
	Frame Type	3	0x10	
	Frame ID	4	0x01	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement). If set to 0, no response is sent.
	64-bit Destination Address	MSB 5	0x00	Set to the 64-bit address of the destination device. The following addresses are also supported: 0x0000000000000000 - Reserved 64-bit address for the coordinator 0x000000000000FFFF - Broadcast address
		6	0x13	
		7	0xA2	
		8	0x00	
		9	0x40	
		10	0x0A	
		11	0x01	
	16-bit Destination Network Address	LSB 12	0x27	Set to the 16-bit address of the destination device, if known. Set to 0xFFFE if the address is unknown, or if sending a broadcast.
		MSB 13	0xFF	
	Broadcast Radius	LSB 14	0xFE	Sets maximum number of hops a broadcast transmission can occur. If set to 0, the broadcast radius will be set to the maximum hops value.
		15	0x00	
	Options			Bitfield of supported transmission options. Supported values include the following: 0x01 - Disable ACK 0x20 - Enable APS encryption (if EE=1) 0x40 - Use the extended transmission timeout for this destination Enabling APS encryption decreases the maximum number of RF payload bytes by 4 (below the value reported by NP). Setting the extended timeout bit causes the stack to set the extended transmission timeout for the destination address. (See chapter 4.) All unused and unsupported bits must be set to 0.
		16	0x00	
	RF Data	17	0x54	Data that is sent to the destination device
		18	0x78	
		19	0x44	
		20	0x61	
		21	0x74	
		22	0x61	
		23	0x30	
	Checksum	24	0x41	
		25	0x13	

Figura 12 - XBee® API Packet (Million Bitz, n.d.)

O módulo XBee® opera em dois modos: API e AT. No modo API a comunicação é feita pelos *packets* mencionados; em modo AT (transparente), o módulo simplesmente retransmite dados

seriais, identificados pelo endereço de 64 bits. Podemos comparar os dois modos conforme a Tabela 10.

Tabela 10 - Comparação entre modo API e AT

Modo API	<ul style="list-style-type: none">• Amostras de I/O. Essa função permite que um módulo XBee® receba dados de I/O de um ou mais XBees® remotos.• <i>Acknowledgement</i> (ACK) e retentativas. Quando enviando um <i>packet</i>, o transmissor recebe um ACK, indicando que o pacote foi entregue com sucesso. O mesmo irá retransmitir o pacote se um ACK não for recebido.• Pacotes recebidos (RX), contém o endereço do transmissor.• Um modulo remoto pode ser configurado para trabalhar em modo AT.• Possibilidade de enviar pacotes em modo <i>BROADCAST</i>.• Possibilidade de obter o RSSI (força do sinal) de cada pacote recebido.• Pacotes incluem um byte de <i>CHECKSUM</i>, para assegurar a integridade dos dados.• Possibilidade de trabalhar com ZigBee <i>endpoints</i>, <i>cluster IDs</i> e <i>profile IDs</i> (XBee® Série 2).
Modo AT	<ul style="list-style-type: none">• Simples.• Compatível com qualquer dispositivo que trabalhe com comunicação serial.• Usado primariamente para comunicação ponto a ponto entre dois módulos XBee®.

3.3.3 Relógio de tempo real

Na intenção de cortar custos, o Raspberry Pi®, diferentemente de um PC, não possui um módulo de relógio de tempo real. Sua hora local é atualizada por um servidor NTP⁶, com a necessidade de que em cada reinício do sistema uma conexão com a internet esteja disponível.

⁶ http://en.wikipedia.org/wiki/Network_Time_Protocol

3.4.1 Unidade de processamento (MCU)

O microcontrolador Atmel® AVR foi inicialmente escolhido graças à sua flexibilidade, bem como a oportunidade de aprendizagem de construção e codificação de tal dispositivo a partir do zero. No entanto, percebeu-se que o tempo e esforço de configuração e codificação deste MCU não justifica o mesmo resultado final que poderia ser mais facilmente alcançado utilizando-se a linguagem Arduino⁸ e seu *Bootloader*⁹. Um rápido estudo foi realizado para garantir que este MCU seria capaz de fornecer todos os requisitos do projeto.

A primeira versão do Node é relativamente simples, sendo necessárias apenas duas portas de comunicação serial: uma para o controle do IC de sensoriamento de energia e outra para o controle do módulo de comunicação sem fio. Por essa razão, uma versão mais simples do microcontrolador foi escolhida. Mesmo na segunda versão, quando um relé e um piezo sonoro serão adicionados, o MCU escolhido ainda será suficiente, visto que possui diversas GPIO digitais ainda não utilizadas.

O modelo escolhido foi o atmega328P, no encapsulamento TQFP. Abaixo são mostradas suas características.

Tabela 11 - Especificações atmega328P

Especificação	Valor
Fabricante:	Atmel®
Núcleo:	AVR
Largura do barramento de dados:	8 bits
Frequência de operação máxima:	20MHz
Tamanho da memória do programa:	32kB
Tamanho RAM dos dados:	2kB
Canais A/D disponíveis:	6
Tamanho de bit A/D:	10 bits
Tensão de alimentação operacional:	1.8V até 5.5V
Tipo de dados Ram:	SRAM
Tamanho ROM dos dados:	1kB
Tipo de dados Rom:	EEPROM
Tipo de interface:	I2C, SPI, UART
Número de I/Os:	23 I/O

⁸ <http://arduino.cc/> - <http://arduino.cc/en/Hacking/BuildProcess>

⁹ <http://arduino.cc/en/Hacking/Bootloader>

Como o modelo escolhido possui apenas uma porta serial, foi necessário fazer a emulação de uma segunda, via software, utilizando a biblioteca nativa da linguagem Arduino. Antes dos testes serem feitos, essa necessidade de emulação gerou dúvidas sobre a estabilidade da mesma, porém, depois de diversos testes e pesquisas, nenhum problema foi encontrado, com a porta virtual se comportando exatamente como a nativa, sem nenhuma perda significativa de performance. Entretanto, para garantir a maior estabilidade possível, a porta serial emulada foi utilizada na comunicação com o CI sensor de energia, pois sua velocidade de comunicação é menor em comparação com o módulo de comunicação sem fio, 9600 *baud* e 38400 *baud* respectivamente.

O circuito necessário para o funcionamento do atmega328P é simples e enxuto e é mostrado na Figura 15.

Para que não houvesse a necessidade de um conversor de nível lógico para a comunicação entre o atmega328P, o módulo XBee® e o sensor de energia, será necessário diminuir a frequência de operação dos normais 16MHz, onde a tensão de funcionamento é de 5V, para 8MHz, podendo assim operar em 3.3V, ou seja, o mesmo nível dos outros componentes utilizados. Testes serão necessários para garantir que essa diminuição na frequência, que diminui o poder de processamento do microcontrolador, não resultará em atrasos no processamento do Node.

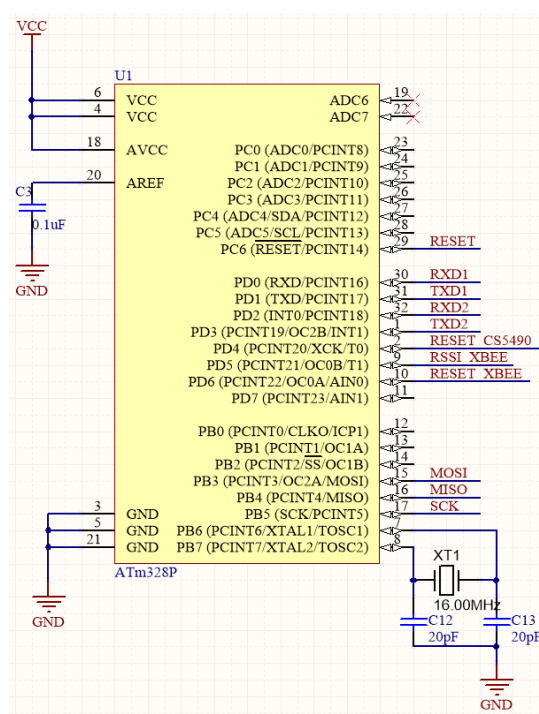


Figura 15 - Circuito mínimo para o CI atmega328P

3.4.2 Sensor de medição de consumo e qualidade da energia

Um dispositivo é necessário para medir com precisão os quilowatts-hora (*kWh*) faturáveis de energia utilizada pelos aparelhos em uma residência. O IC de medição de energia escolhido foi o CS5490, produzido pela Cirrus Logic.

Devido à prática comum na indústria, a unidade de medição de energia elétrica será o quilowatt-hora (*kWh*), em vez da unidade do SI joule (*J*). A taxa de conversão é indicada a seguir.

$$1 \text{ kWh} = 3.6 \times 10^6 \text{ J}$$

Equação 1 - Conversão entre kWh e Joule

Medidores de energia comerciais foram considerados, tais como o CRD5490¹⁰, também da Cirrus Logic, mas, o alto custo, a necessidade de modificações para a inclusão de comunicação sem fio e tamanho acima do desejado impediram sua utilização. Portanto, a seguinte decisão foi tomada: projetar e construir o sensor. Extensa pesquisa foi feita sobre o assunto para produzir um projeto de fácil utilização, acessível e seguro.

Para o projeto, apenas instalações elétricas de fase única foram consideradas, visto que são a grande maioria nas residências. Apenas cargas resistivas foram consideradas nos cálculos abaixo, sendo então o cosseno do ângulo de fase igual a 1 ($\cos 0^\circ = 1$). O sistema final será capaz de calcular a potência não só de cargas puramente resistivas, como também com componentes indutivos e capacitivos.

$$P = \frac{1}{2} \cdot V_p \cdot I_p \cdot \cos(\theta) \rightarrow P = V_{RMS} \cdot I_{RMS} \cdot \cos(\theta)$$

Equação 2 – Cálculo de potência

Para encontrar a quantidade de energia consumida, é feita a integração no tempo da potência.

$$\begin{aligned} E &= \int_0^t P(t) dt \\ &= P \cdot t \end{aligned}$$

Equação 3 - Cálculo da energia consumida

Com isto em mente, o trabalho pôde ser iniciado com o projeto e construção do sensor de energia. Outras alternativas além do CI da Cirrus Logic foram consideradas e comparadas. O CS5490 foi, a partir de pesquisas em diversos fóruns sobre o assunto e conversas com professores, considerado a opção mais interessante. Com um CI exclusivo para a medição de energia, considerável peso computacional pôde ser aliviado do MCU, sem a necessidade de cálculos matemáticos complexos, apenas leitura de registradores via interface serial. Com

¹⁰ http://www.cirrus.com/en/pubs/rdDatasheet/CRD5490-Z_RD1.pdf

a diminuição da frequência de operação do MCU, necessária para o funcionamento em 3.3V, o tempo de computação de cada *loop* foi uma preocupação constante. Para que a carga computacional no atmega328P fosse a menor possível todos os cálculos não obrigatórios para a comunicação com o CI de sensoriamento foram transferidos para a Central, que possui um poder de processamento muito superior ao do microcontrolador.

Para o projeto prosseguir, algumas decisões precisaram ser tomadas. O CS5490 dá ao usuário a opção de como a energia será medida. Os métodos disponíveis de detecção de corrente são: transformador de corrente (TC), resistor *shunt* e Bobina de Rogowski. Dos três, o TC é comumente usado, pois fornece boas leituras e é mais seguro para trabalhar, quando se trata de exposição de fios desencapados. Como um dos objetivos do projeto era de que o sensor fosse o menor possível, o método do resistor *shunt* foi escolhido. Apesar de mais invasivo e potencialmente menos seguro, precauções foram tomadas levando-se isso em consideração, e, a potencial diminuição das dimensões do sistema como um todo teve maior relevância na escolha.

Ambas as entradas I_{IN} e V_{IN} do CI podem receber uma tensão máxima de 250 mV_{AC} (Cirrus Logic, 2013), com a possibilidade de operação em modo de alto ganho de $50\times$, com sua amplitude reduzida para 35 mV_{AC} . Tanto o valor do resistor R_{shunt} quanto dos resistores do divisor de tensão foram escolhidos para utilizar praticamente toda a amplitude de entrada do CI, visando alcançar a melhor precisão possível.

Para manter a menor dissipação de energia possível, um resistor *shunt* em conjunto com o modo de alto ganho foi utilizado. Aplicando a lei de Ohm para a amplitude desejada, podemos ver pela equação 4, que o valor calculado para o resistor é de $2.33\text{ m}\Omega$. Para uma margem de 85%, o valor do resistor foi reduzido para $R_{shunt} = 85\% \cdot 2.33\text{ m}\Omega \cong 2\text{ m}\Omega$. A especificação do *shunt* utilizado foi de duas vezes a potência dissipada em carga máxima, que em 15A, é igual a $2W$. Como pode ser visto no circuito da Figura 16, o CI requer um divisor de tensão nas entradas V_{IN+} e V_{IN-} , com o valor R_1 escolhido, baseado na equação 4. O valor de R_2 foi selecionado previamente em $1\text{ k}\Omega$. Uma margem de 115% foi considerada na escolha do resistor R_1 , tendo seu valor final dividido em quatro resistores de $422\text{ k}\Omega$. Todos os cálculos são baseados em uma tensão de entrada de até 250 V_{RMS} . É importante notar que o CS5490 suporta tensões acima de 250 mV_{AC} (Cirrus Logic, 2013), porém, esses valores são mapeados para o valor máximo da entrada.

$$R_{shunt} = \frac{V_{saida}}{I_{shunt}} = \frac{35\text{ mV}}{15\text{ A}} = 2.33\text{ m}\Omega$$

Equação 4 - Cálculo do resistor shunt

$$R_1 = R_2 \cdot \left(\frac{V_{entrada}}{V_{saida}} - 1 \right) \rightarrow R_1 = 1 \text{ k}\Omega \cdot \left(\frac{250 \text{ V}}{176 \text{ mV}} - 1 \right) \rightarrow R_1 = 1.42 \text{ M}\Omega$$

Equação 5 - Tensão de entrada no CI

O uso de um divisor de tensão conectado entre a linha de fase e neutro de um circuito de alta tensão gerou algumas preocupações. A existência da possibilidade de um curto nessa conexão foi levada em conta, e algumas medidas de segurança precisarão ser tomadas no produto final, como a vedação do revestimento plástico utilizando soldagem por ultrassom.

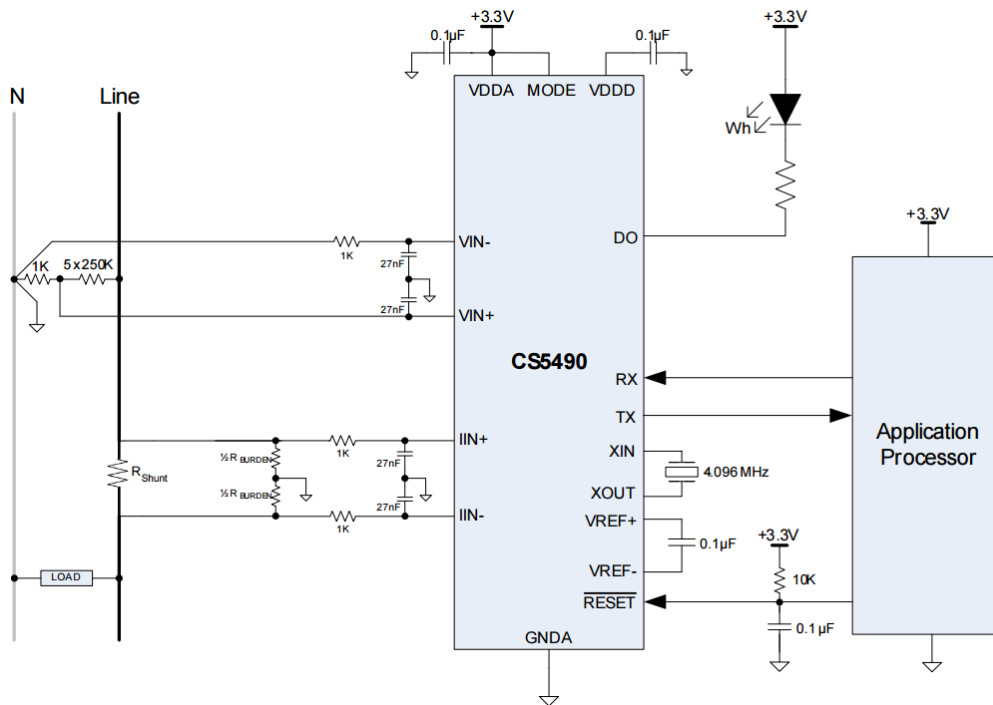


Figura 16 - Circuito CS5490 (Cirrus Logic, 2013)

Para um correto funcionamento do CI CS5490, é necessária sua prévia calibração, processo mostrado na seção 5.1.

3.4.3 Comunicação sem fio

A comunicação sem fio no Node é feita com o mesmo módulo XBee® utilizado na Central, com diferenças apenas na configuração. Na Central, como já foi dito, o módulo XBee® funciona em modo Administrador, sendo este responsável pela criação, configuração e gerenciamento da rede Zigbee. Os Nodes são configurados para trabalhar em modo Roteador, comunicando-se diretamente com o Administrador. Módulos XBee em modo *Endpoint* não foram utilizados no sistema

Pela função de redirecionamento de mensagens do protocolo Zigbee, pode haver comunicação roteadores. Isso acontece quando não é possível que um Node se comunique diretamente com a Central, sendo feito o roteamento das mensagens por um ou mais Nodes. Todo esse processo é feito de forma transparente, sem aumento significativo no tempo de entrega das mensagens. Para pacotes de trinta bytes, utilizando três redirecionamentos (Roteador Origem / primeiro redirecionamento → Roteador do segundo redirecionamento → roteador do terceiro redirecionamento → Coordenador), esse tempo é de, aproximadamente, 200 milissegundos (Piyare & Lee, 2013), valor aceitável para a aplicação.

4 APRESENTAÇÃO DO PROTÓTIPO

4.1 Módulos sensores

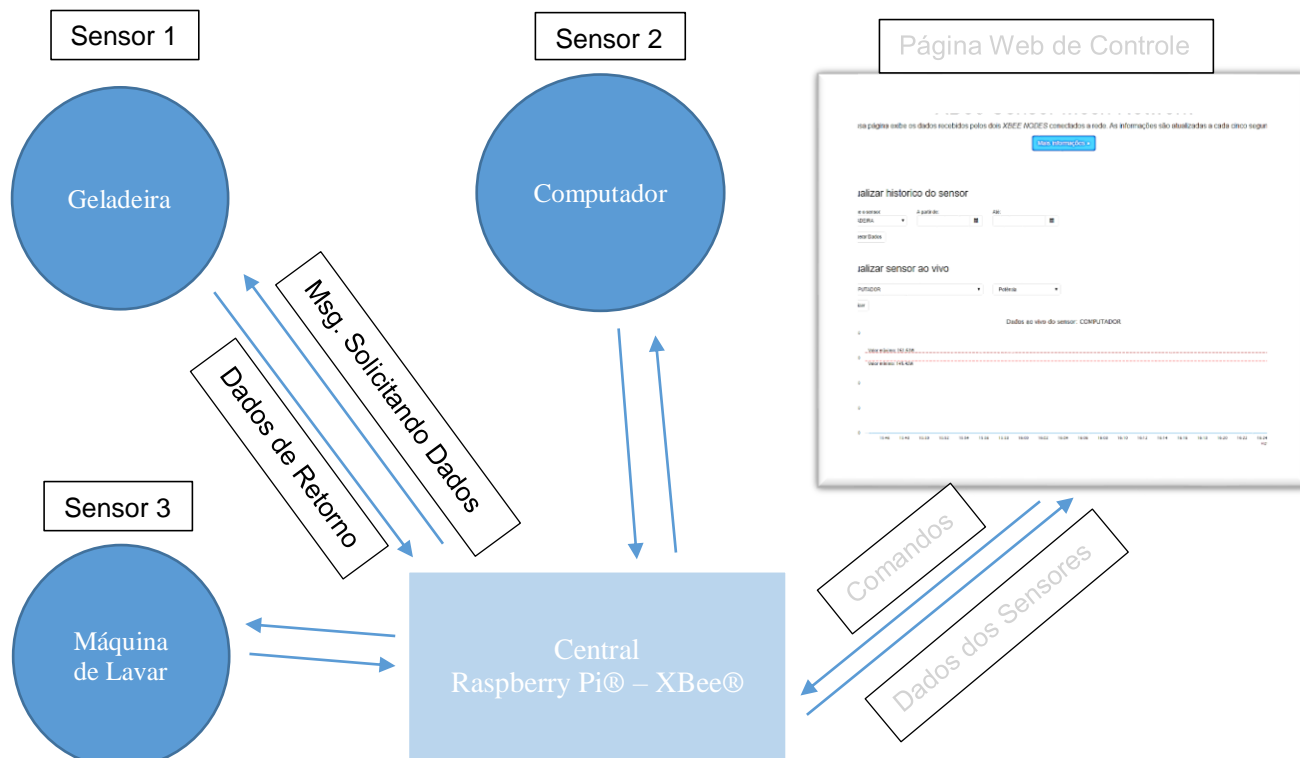


Figura 17 - Esquemático enfatizando os módulos sensores

Cada Node é composto por um microcontrolador, um sensor de energia e o módulo de comunicação sem fio.

Como prova de conceito e para facilitar os testes durante o desenvolvimento, o sensor de potência foi separado do microcontrolador e do módulo de comunicação sem fio. Isto foi feito para que o Node pudesse ser conectado ao computador enquanto o sensor estivesse ligado à rede. Se essa separação não tivesse sido levada em conta, ao ser conectado em uma porta usb, o Node teria dois pontos de terra em tensões diferentes, o que ocasionaria sérios problemas. Para que esse desacoplamento fosse seguro, três optoacopladores foram utilizados para a conexão das linhas de transferência e recebimento serial, assim como para o reset via software do CI CS5490.

Nas **Erro! Fonte de referência não encontrada.** eFigura 19 é mostrada parte do Node esquematizado e desenvolvido para o projeto, não estando inclusos na imagem o microcontrolador e modulo XBee®. Nela é mostrado apenas o módulo sensor, que contém o CI CS5490 e sua fonte de alimentação. A criação da placa de sensoriamento foi feita após longo estudo do *datasheet* dos CIs CS5490 (sensor de potência) e LM2594 (fonte chaveada), bem como da placa de avaliação do CI sensor.

A Figura 20 mostra a placa do microcontrolador e módulo de comunicação XBee®, sendo a conexão entre os dois feita por um *shield*¹¹ para Arduino®.

Para o funcionamento do Node, uma biblioteca¹² específica para comunicação com o módulo XBee® foi utilizada. Essa biblioteca foi desenvolvida há mais de seis anos e apresentou-se de forma robusta em todos os testes efetuados. Para a comunicação com o CI CS5490 a criação de uma lista de métodos foi necessária, visto que nenhuma biblioteca para Arduino foi encontrada. Os métodos são utilizados principalmente na leitura e gravação de registradores, com alguns deles mais específicos, onde cálculos e manipulações binárias são necessários, como na seleção de uma velocidade diferente de comunicação serial ou na verificação de que o CI finalizou mais uma sequência de cálculos.

Levando-se em conta diferenças no processo de manufatura, o CI CS5490 precisa ser calibrado para seu correto funcionamento. Este processo de calibração é feito apenas uma vez, onde as variáveis de calibração são calculadas e é explanado na próxima seção.

O módulo XBee® série 2 pode enviar por mensagem aproximadamente (o valor varia de acordo com as configurações escolhidas), 80 Bytes de dados, sem contar cabeçalhos de mensagem, *checksum* e outros valores obrigatórios. Porém, quanto maior o pacote enviado maiores são as chances de ocorrerem perdas de dados. Pensando nisso, os dados escolhidos para o envio foram: potência, tensão, corrente, fator de potência e temperatura, totalizando 15 bytes.

Algumas medidas podem ser tomadas para diminuir a quantidade de dados enviados por mensagem, como a remoção da potência, sendo ela calculada usando os valores de tensão, corrente e fator de potência, ou até a compactação dos dados, entretanto, ambos os casos resultariam em uma maior carga de processamento.

Como a placa de sensoriamento precisa ser conectada diretamente na tensão de rede, alguns cuidados foram tomados, como uma maior atenção na distância de separação entre traços de cobre que tivessem uma alta diferença de tensão, como por exemplo, a rede terra e a rede onde o resistor *shunt* está conectado. Para o cálculo de distâncias seguras entre traços, foi utilizado uma calculadora¹³ feita para esse propósito, seguindo métricas especificadas na segunda edição do documento UL 60950-1¹⁴.

No escopo deste projeto, o módulo sensor foi programado para receber três tipos de mensagens vindas da Central: a primeira solicita o envio de uma nova lista de dados; a

¹¹ http://www.robotizando.com.br/shields_index.php

¹² <https://github.com/andrewrapp/XBee@-arduino>

¹³ <http://www.smeps.us/pcbtracespacing.html>

¹⁴ http://www.ul.com/global/documents/offerings/industries/hightech/informationtechnology/new/60950-1_2ndEd_Analysis_rev_May%2021%202010.pdf

segunda fornece o endereço de 64 bits do coordenador e a terceira solicita que o Node altere seu nome.

Quando uma nova lista de dados é solicitada, o microcontrolador inicialmente checa com o CI CS5490 se um novo grupo de medições já está disponível, e em seguida, se prontas, lê todos os registradores responsáveis por armazenar cada uma das medições.

Cada medição é armazenada em três bytes e é transmitida com o bit menos significativo primeiro (*LSB first*). Para que os dados sejam enviados na sequência correta são necessárias manipulações binárias. A leitura e manipulação do registrador que armazena os dados de corrente pode ser visto no código abaixo.

```
uint32_t data_temp = SP_seleciona_pagina_e_le_registrador(16, 6);
Buffer[2] = (data_temp & 0x000000ff);
Buffer[1] = (data_temp & 0x0000ff00) >> 8;
Buffer[0] = (data_temp & 0x00ff0000) >> 16;
```

Pelo fato do CI CS5490 não possuir memória não-volátil, é necessário que, a cada reinício, os valores de calibração sejam gravados em seus respectivos registradores. Após essa gravação, o registrador de *checksum* dos registradores é lido e comparado com o valor já registrado. Isso é feito para que qualquer problema na inicialização do sensor possa ser verificado e corrigido. Caso os valores de *checksum* (lido e em memória) não sejam iguais, o CI é reiniciado via *software* pelo microcontrolador e o processo é feito novamente.

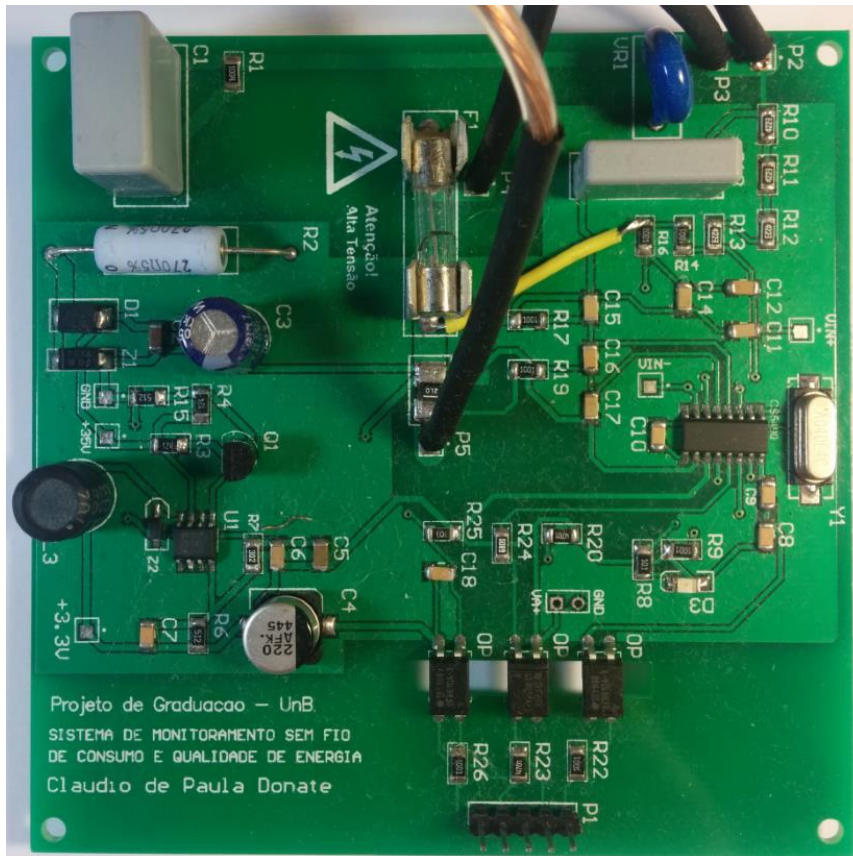


Figura 18 – Placa de sensoriamento montada

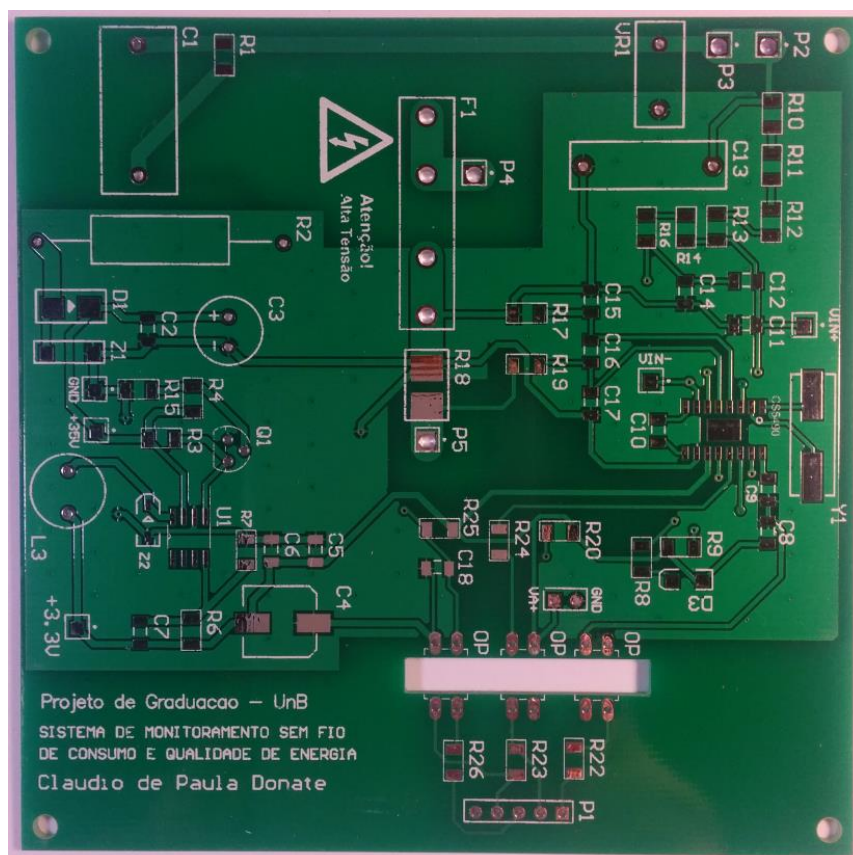


Figura 19 - Placa de sensoriamento

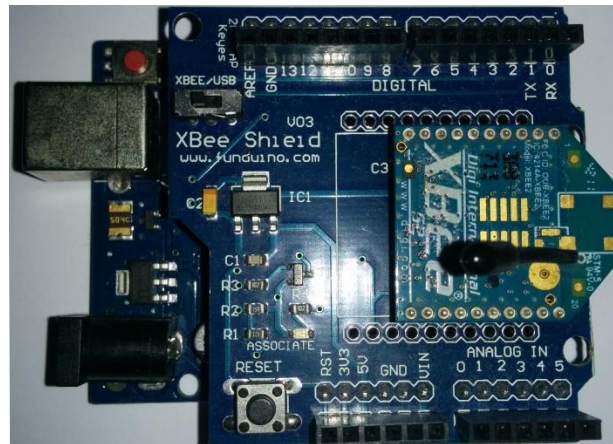


Figura 20 - Módulo microcontrolador e de comunicação

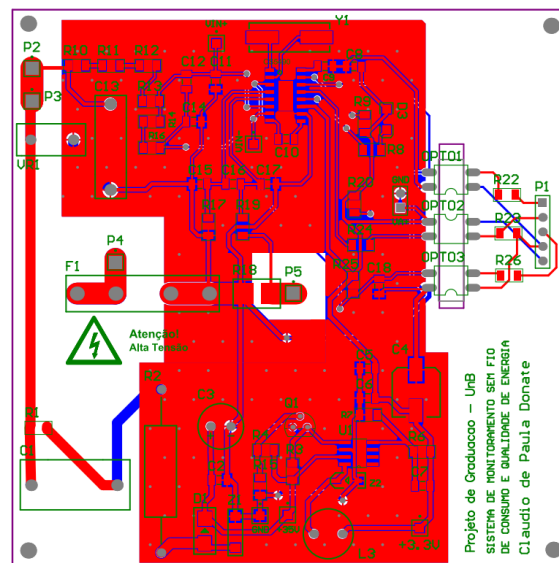


Figura 21 - Circuito Sensor – Topo

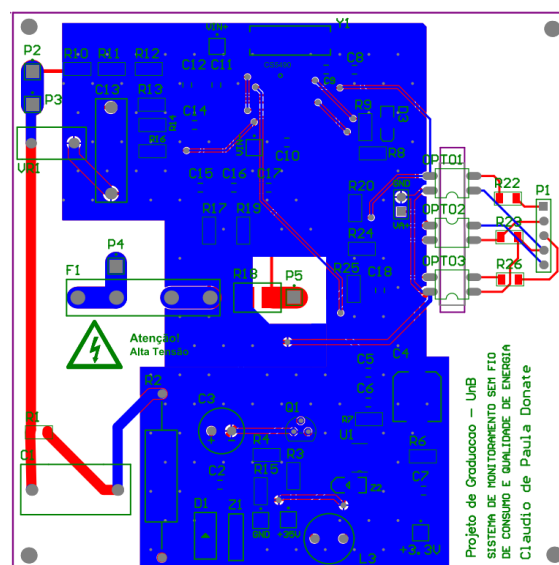


Figura 22 - Circuito Sensor – Fundo

4.2 Central de controle

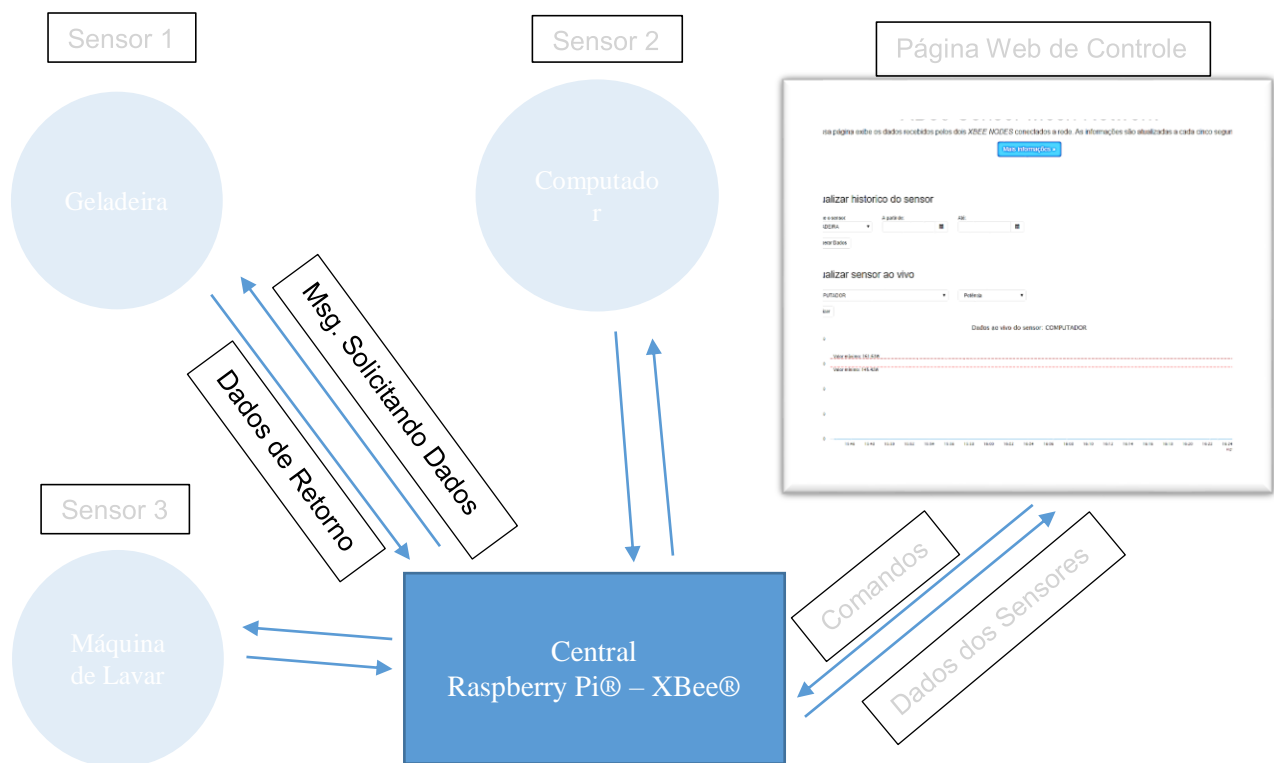


Figura 23 - Esquemático enfatizando a Central de processamento

A Central é composta por um microcomputador Raspberry Pi®, um módulo de comunicação XBee® e um relógio de tempo real (RTC). Para fins de desenvolvimento, testes e depuração, na primeira etapa do projeto, o módulo RTC não foi implementado. Com a constante disponibilidade de conexão com a internet, o relógio interno do *Kernel* Linux era sempre atualizado em tempo de *boot*. Nessa fase inicial, também um módulo XBee® é conectado externamente, via usb. A necessidade de mudanças constantes na configuração do módulo de comunicação sem fio fez que com esse modo de trabalho fosse mais adequado.

A placa de circuitos contendo os módulos XBee® e RTC foi desenvolvida e é mostrada na Figura 24, com o esquemático do circuito, Figura 25, e a placa de circuitos física, Figura 26.

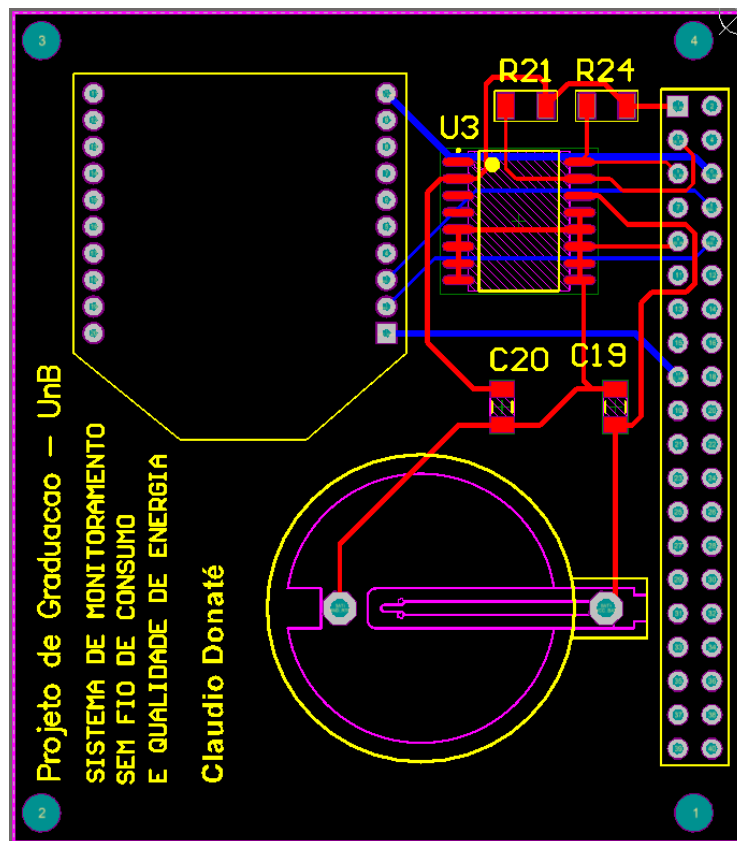


Figura 24 - Circuito Xbee®/RTC Central

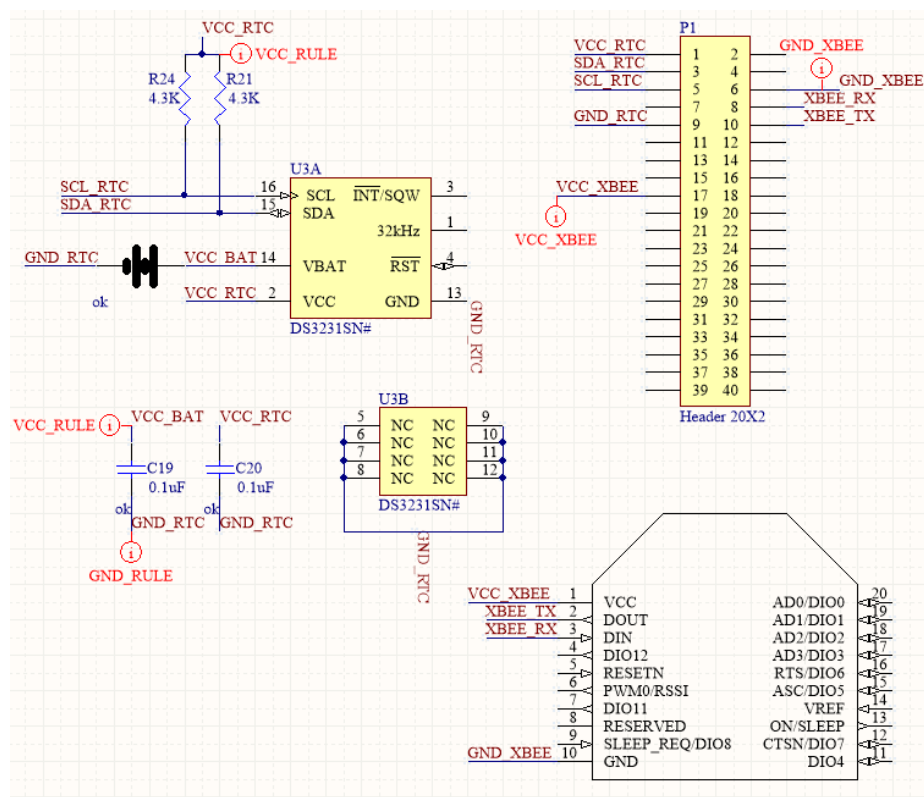


Figura 25 - Esquemático Xbee® / RTC Central

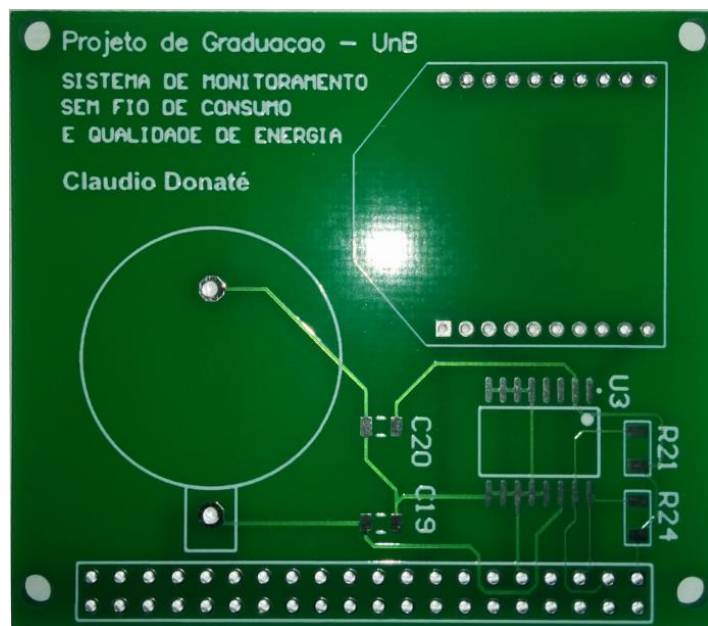


Figura 26 - Placa de circuito – Central de processamento

É possível ver que a placa é simples, apenas conecta os módulos XBee® e RTC aos pinos GPIO do Raspberry Pi®. Foram colocadas quatro aberturas, uma em cada um dos vértices, para que a placa pudesse ser aparafusada ao microcomputador, evitando que vibrações causassem algum mal contato.

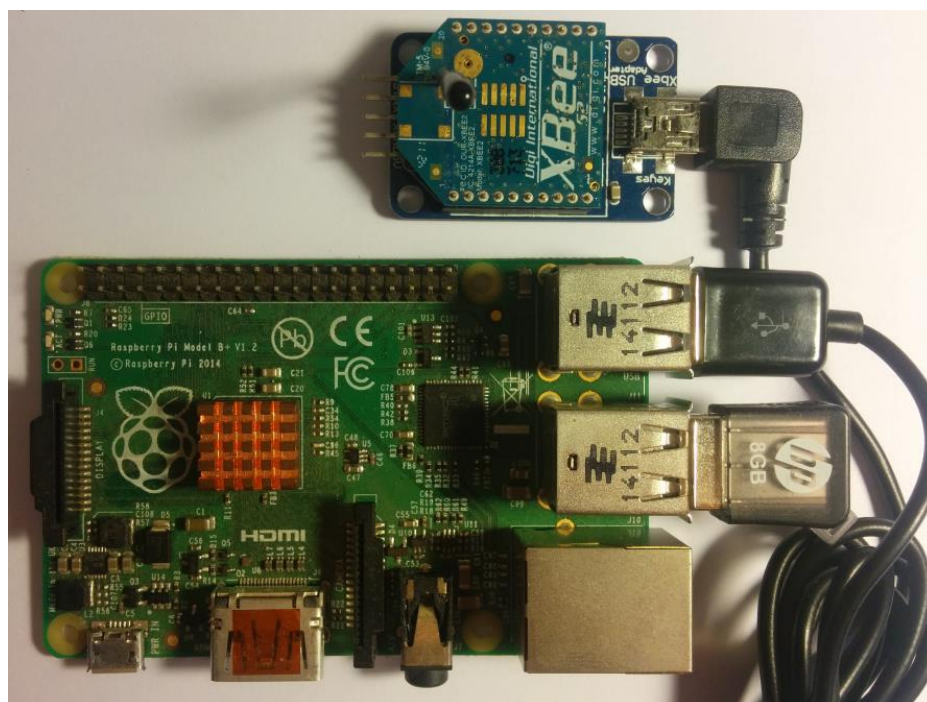


Figura 27 – Central de Processamento

Um dos pontos observados no projeto da Central foi que o Raspberry Pi® possui como memória interna um cartão microSD. Na pesquisa inicial, descobriu-se que o mesmo, em ambientes com um número excessivo de gravações, pode, depois de um curto período de tempo, sofrer problemas com o limite físico do número de gravações que essa tecnologia possui. Mesmo com a utilização de políticas de *wear-leveling*¹⁵, onde as gravações são espalhadas por todo o cartão, entendemos ser mais prudente utilizar um pen-drive em conjunto com o cartão microSD. Essa forma de memória externa foi utilizada apenas para o banco de dados de gravação dos dados provenientes dos Nodes. O banco de dados foi copiado do cartão interno e todos os arquivos de configuração internos modificados, para apontarem para o pen-drive, que é automaticamente montado¹⁶ e configurado em cada reinício do sistema. Com uma capacidade de 8GB, sua expectativa de vida não influenciaria negativamente o tempo de vida útil da Central.

A Central tem diversas funções: controle do módulo XBee® e toda a lógica de conexão/desconexão e controle dos Nodes; funcionar como servidor HTTP para a página de controle e visualização de dados; controlar as leituras e gravações no banco de dados; controlar o servidor de *websocket* e transformar os dados binários dos sensores em valores úteis para o usuário.

4.2.1 Script de controle do módulo XBee®

O *script* de controle do módulo XBee®, chamado de XBee_handler.py, foi escrito em Python, linguagem já familiarizada, que uma robusta biblioteca de controle do módulo XBee®, o que acelerou muito o desenvolvimento. Esse script é utilizado pelo servidor de *websocket*, que, quando iniciado, cria um objeto XBeeObject, que dá início ao processo de comunicação com os módulos sensores, além de dar acesso às funções necessárias do script XBee_handler.py.

Antes da inicialização do servidor de *websocket*, no momento em que o *script* XBee_handler é importado, tanto a porta serial, quanto o objeto ZigBee (obrigatório para a gerenciamento do módulo de controle XBee®) são inicializados. Assim que o objeto XBeeObject é instanciado, o comando *Node Discover*¹⁷ (ND) é enviado em modo *broadcast* pelo coordenador. Este comando solicita a todos os Nodes conectados que enviem uma mensagem ao coordenador com suas informações básicas, como seu nome identificador e endereço. Estas informações são usadas para construir um dicionário com dados de cada um dos Nodes conectados. Ele será usado para gerenciar todo o processo de solicitação de

¹⁵ https://en.wikipedia.org/wiki/Wear_leveling

¹⁶ [https://pt.wikipedia.org/wiki/Mount_\(Unix\)](https://pt.wikipedia.org/wiki/Mount_(Unix))

¹⁷ http://examples.digi.com/wp-content/uploads/2012/07/XBee_ZB_ZigBee_AT_Commands.pdf

dados, recebimento e controle de dispositivos conectados. Os dados são: endereço de 64 bits do Node, identificador, o número de tentativas de comunicação entre o coordenador e Node que não foram completadas com sucesso, os dados recebidos e se os dados atualmente ali presentes já foram lidos pelo servidor de websocket.

Fazendo uso de um sistema de agendamento de tarefas, a cada cinco segundos é feito o envio de uma mensagem em *broadcast* solicitando dados dos Nodes conectados. Este sistema de agendamento faz uso de técnicas de *multithread*, o que faz com que cada envio seja feito independentemente de outros processos que estejam acontecendo. Este intervalo de cinco segundos é o tempo mínimo entre solicitações de dados e é mantido para, no máximo, dez módulos sensores conectados. A partir do momento em que o número de Nodes conectados ultrapasse esse valor, o intervalo é estendido para dez segundos entre mensagens. Isto é feito para que a Central de processamento tenha tempo hábil de processar todas os dados recebidos, diminuindo assim as chances de que ocorra alguma colisão de mensagens.

Após o envio da mensagem de solicitação de dados, o *script* adiciona, no dicionário mencionado acima, uma unidade ao número de tentativas de comunicação em cada um dos Nodes. Depois de recebida pela Central a resposta do Node, o *script* subtrai uma unidade do número de tentativas. Se esse valor ultrapassar três unidades, o sistema entende que o Node foi desconectado e inicia o processo de descoberta de sensores conectados, feito pelo comando ND. Isto faz com que qualquer Node desconectado seja removido do dicionário em no máximo quinze ou trinta segundos, dependendo do tamanho da rede.

Cada módulo sensor é configurado para que envie uma mensagem ao coordenador assim que entra na rede. Caso a Central receba esta mensagem, o novo Node conectado é adicionado ao dicionário e isto faz com que novos sensores possam ser adicionados à rede a qualquer momento.

Além do gerenciamento de mensagens, o *script* XBee_handler.py administra todas as solicitações do servidor de websocket, que pode solicitar dados ao vivo de um determinado Node, solicitar mudanças de nome ou que dados não sejam gravados no Banco de dados, para que um sensor possa ser mudado de local.

Visando a diminuição da carga de processamento no microcontrolador, os dados são enviados no mesmo formato que são recuperados do CI CS5490. Por essa razão a Central precisa transformar os dados binários em complemento de 2^{18} , com o ponto decimal em diferentes bits, para valores úteis ao usuário. A conversão é feita a cada recebimento de

¹⁸ https://pt.wikipedia.org/wiki/Complemento_para_dois

mensagem e utiliza uma forma de cálculo simplificada. Para os valores de corrente e tensão é utilizada a Equação 5.

$$VALOR_{Decimal} = \frac{1}{2^{24} - 1} \cdot hex2dec(VALOR_{Hexadecimal})$$

Equação 5 - Equação para cálculo da tensão e corrente

Quanto aos demais valores, a mesma equação é utilizada, diferenciando-se apenas pelo valor do expoente no dividendo, que se relaciona diretamente ao número de bits após o ponto binário, ou seja, os valores de temperatura, que podem variar de -128,0°C até 128,0°C, tem seu ponto binário à direita do bit 16, e, assim sendo, o expoente no cálculo seria 16.

4.2.2 Servidor websocket

Para a transferência de dados entre a Central e a página de controle, foi utilizada a tecnologia websocket, que permite comunicação bidirecional por canais *full-duplex* sobre um único soquete TCP (*Transmission Control Protocol*).

O preceito básico para a escolha do servidor de websocket foi a mesma do servidor HTTP (descrita na próxima seção), um sistema leve que fosse pouco taxativo no microcomputador e provesse todas as ferramentas necessárias.

O servidor escolhido foi o *Simple Websocket Server*¹⁹, desenvolvido em Python, que também foi utilizada no script de controle do módulo XBee®, o que facilitou a comunicação e transferência de dados entre os dois.

O script criado para o servidor é usado para desencadear todo o processo da Central, inicializando o script de controle XBee®, assim como o próprio servidor *websocket*. Ao final do processo, o *script* aguarda, na porta escolhida, conexões vindas da página de controle.

Assim que a página de controle é solicitada do servidor HTTP, uma requisição de conexão via websocket é feita ao servidor. Após uma conexão bem-sucedida, o servidor envia à página de controle o nome e endereço dos Nodes conectados atualmente, além de todos os que, em algum momento, estiveram conectados à Central, para que o usuário possa visualizar não só dados ao vivo, como dados do histórico de Nodes conectados.

O usuário, via página de controle, pode solicitar dados ao vivo de qualquer sensor conectado, bem como dados de um sensor em um período de tempo específico. Os dados ao vivo vêm diretamente do script de controle XBee®, diferentemente dos dados de histórico que são buscados no banco de dados.

¹⁹ <https://github.com/dpallot/simple-websocket-server/>

O servidor também trata solicitações de mudança de identificador do Node e pausa na gravação dos dados no Banco de dados. Isso é feito via métodos que compõem o objeto XBeeObject, descrito na seção anterior.

4.2.3 Script de acesso ao banco de dados

Foi necessária a criação de um *script* responsável pela manutenção da conexão ao banco de dados. Durante os testes, verificou-se que o sistema periodicamente perdia essa conexão, o que resultava em um erro não recuperável, e o sistema não mais respondia a requisições.

Para que esse problema fosse superado, um script externo, desenvolvido também em Python, foi criado. Ele recebe todas as requisições de leitura e gravação no banco de dados e, envolto em um código de tratamento de exceções²⁰ *try/catch*, testa a conexão com o banco antes de executar a tarefa selecionada. Verificando-se que ligação não existe mais, ele a restabelece e executa a tarefa.

Outro problema encontrado: no momento de início do sistema (*boot*), o *script* do servidor de websocket, que inicia todos os outros processos, era chamado antes que o banco de dados MySQL finalizasse seu estabelecimento. Isto acarretava erros que encerravam todo o processo. Para superar este obstáculo, outro código de tratamento de exceções foi criado e colocado em um método separado, que é chamado no momento da inicialização do servidor de websocket e testa, repetidamente, se a conexão com o banco de dados está ativa. Só após receber resposta positiva o processo de inicialização pode continuar.

Após tomadas as medidas descritas nesta seção, nenhum problema com o banco de dados foi detectado.

4.2.4 Banco de dados

Para a escolha do banco de dados a ser utilizado, dois fatores foram levados em conta: o consumo de recursos e a facilidade de uso.

Já éramos familiarizados com o banco de dados MySQL, e sabíamos não ser ele o mais leve para utilização em ambientes embarcados. Porém, após pesquisas, foi descoberto que ele poderia ser configurado para trabalhar em modo reduzido, o que diminui muito seu consumo e o deixa propício ao funcionamento no Raspberry Pi®.

Apenas duas tabelas foram necessárias: *sensor_data* e *sensor_history*. A primeira guarda todos os dados recebidos por cada um dos módulos sensores e sua estrutura pode ser vista na Tabela 12.

Tabela 12 - Estrutura da tabela *sensor_data*

²⁰ https://pt.wikipedia.org/wiki/Tratamento_de_exceção

Coluna	Tipo	Dado
<i>id</i>	int(11)	Identificador do registro no banco
<i>sensor_addr</i>	varchar(16)	Endereço de 64 bits do sensor
<i>sensor_name</i>	varchar(100)	Nome do sensor
<i>sensor_current</i>	float(4,2)	Valor de corrente medida
<i>sensor_voltage</i>	float(5,2)	Valor de tensão medida
<i>sensor_power</i>	float(6,2)	Valor de potência medida
<i>sensor_pf</i>	float(3,2)	Valor de fator de potência medida
<i>sensor_temp</i>	float(5,2)	Valor de temperatura medida
<i>sensor_datetime</i>	datetime	Data e hora da medição

A segunda armazena o registro de todos os sensores que se conectaram à central em algum momento. Sua estrutura pode ser vista na Tabela 13.

Tabela 13 - Estrutura da tabela *sensor_history*

Coluna	Tipo	Dado
<i>id</i>	int(11)	Identificador do registro no banco
<i>sensor_addr</i>	varchar(16)	Endereço de 64 bits do sensor
<i>sensor_name</i>	varchar(100)	Nome do sensor

4.2.5 Servidor HTTP

A escolha do servidor HTTP foi baseada em um único critério, a saber, o quão taxativo o mesmo seria no microcomputador. Como é necessário apenas servir arquivos estáticos, até o servidor mais simples seria suficiente, onde apenas as funcionalidades básicas seriam utilizadas. Com o embasamento colhido em diversos artigos, onde testes específicos para o Raspberry Pi® foram feitos, o servidor escolhido foi o Lighttpd²¹, que é extremamente leve e possui todas as funcionalidades necessárias.

²¹ <https://www.lighttpd.net/>

4.3 Interface de Controle

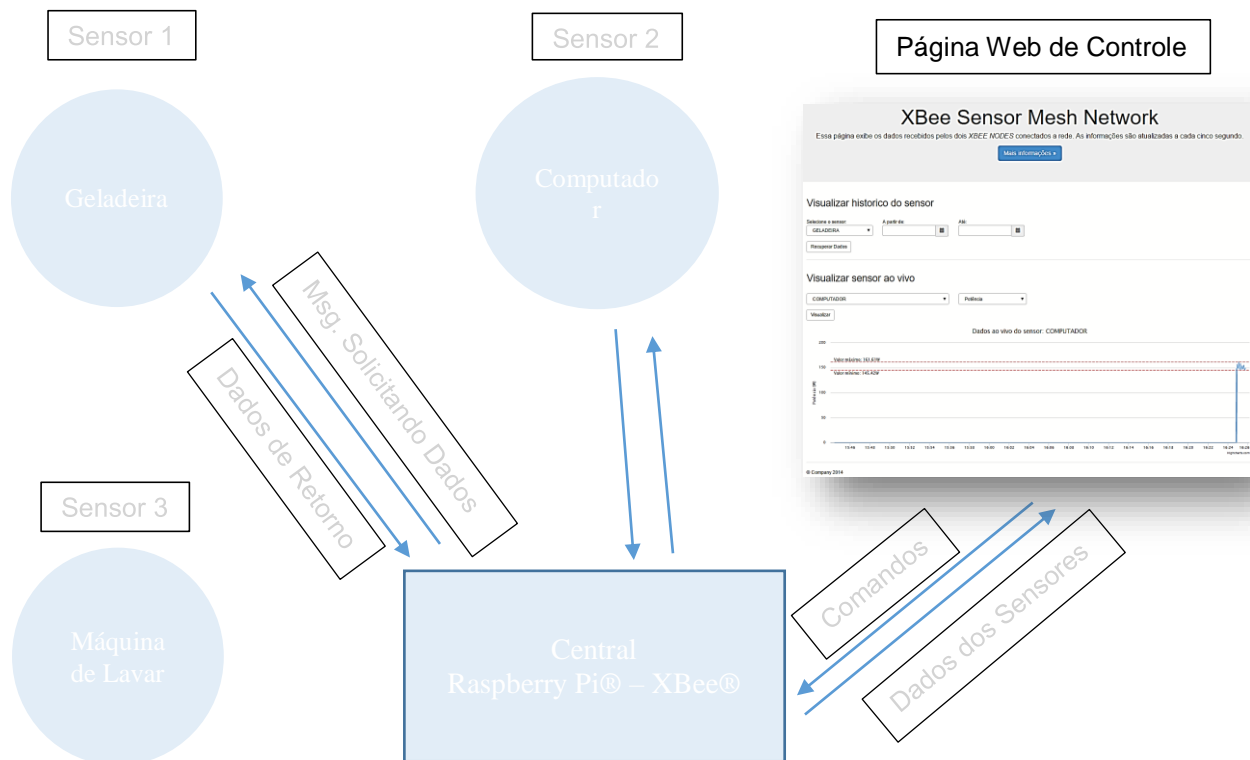


Figura 28 - Esquemático enfatizando a página de controle

A página de controle foi feita da forma mais simples possível. Para tanto, foi a utilização do *framework Twitter Bootstrap*, que acelerou seu desenvolvimento.

Assim que a página é aberta, uma conexão websocket é estabelecida. Duas mensagens são enviadas pelo servidor: a primeira contendo a lista com todos os Nodes conectados e a segunda com as informações de todos os Nodes que em algum momento se conectaram com a Central.

Quando dados de histórico são selecionados, a página de controle envia ao servidor, em formato JSON²², o intervalo de tempo e as informações do Node desejado. O servidor, por sua vez, busca no Banco todos os dados que estão dentro deste intervalo e os retorna, também em formato JSON, para que a página crie o gráfico do intervalo desejado. Exemplos destas mensagens podem ser vistos abaixo:

- Solicitação de dados de histórico:

²² <https://pt.wikipedia.org/wiki/JSON>

```
{instruction: "recupDados", node: "GELADEIRA", init: "2016-01-04 00:00:00", end: "2016-01-04 06:00:00"}
```

- Resposta do servidor de websocket:

```
{instruction: "graphDados", dados: Array[3094]}
```

Na opção de visualizar os dados ao vivo, a página envia para o servidor, a cada cinco segundos, os dados do Node desejado, onde o servidor retorna com as medições atuais recebidas do Node solicitado. Este dado é então adicionado ao gráfico. Exemplos destas mensagens podem ser vistos abaixo:

- Solicitação de dados ao vivo:

```
{"instruction": "getLiveData", "nodeAddr": "0013a20040869ae1"}
```

- Resposta do servidor de websocket:

```
{"instruction": "liveData", "data": {"status": true, "temp": 27.965270655488002, "power": 151.03537319980347, "datetime": "2016-01-31 11:31:29", "current": 0.8423352743854934, "pf": 0.8386035965093321, "voltage": 213.81595515787794}}
```

Todos os gráficos podem apresentar, como dado do eixo y, potência, tensão ou corrente, além de apresentar para cada ponto do eixo x essas três informações que são acompanhadas ainda do fator de potência e da temperatura do Node.

Pela página de controle é possível também modificar o nome dos sensores conectados e pausar a gravação de dados de um determinado Node no banco de dados. Essas funções foram adicionadas para que a troca de local de um Node não gerasse dados inconsistentes.

A Figura 29 mostra os dados ao vivo do Node Computador, onde a potência foi selecionada como eixo y. No canto inferior esquerdo, podem também ser vistas todas as informações disponíveis para o ponto do gráfico selecionado. A Figura 30 mostra os dados de histórico do sensor Geladeira capturados em um período de quatorze horas. Já a Figura 31 apresenta todos os Nodes atualmente conectados, com seus nomes em campos editáveis e seus respectivos status, tendo esses dois valores possíveis: GRAVANDO ou DESATIVADO.

XBee Sensor Mesh Network

Essa página exibe os dados recebidos pelos dois XBEE NODES conectados a rede. As informações são atualizadas a cada cinco segundo.

[Mais informações »](#)

Visualizar historico do sensor

Selezione o sensor:
 A partir de:
 Até:

[Recuperar Dados](#)

Visualizar sensor ao vivo

COMPUTADOR
 Potência

[Visualizar](#)

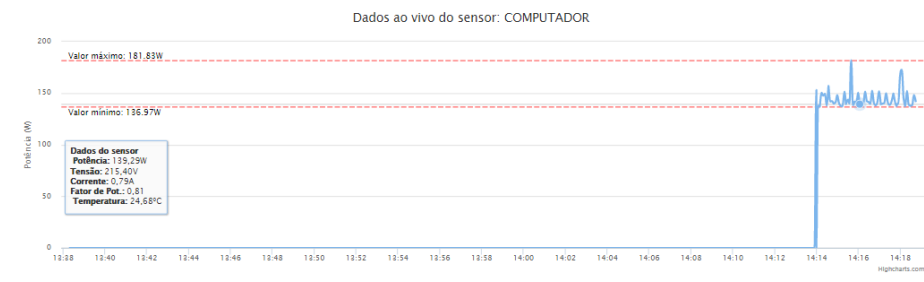


Figura 29 - Página de controle mostrando dados ao vivo

XBee Sensor Mesh Network

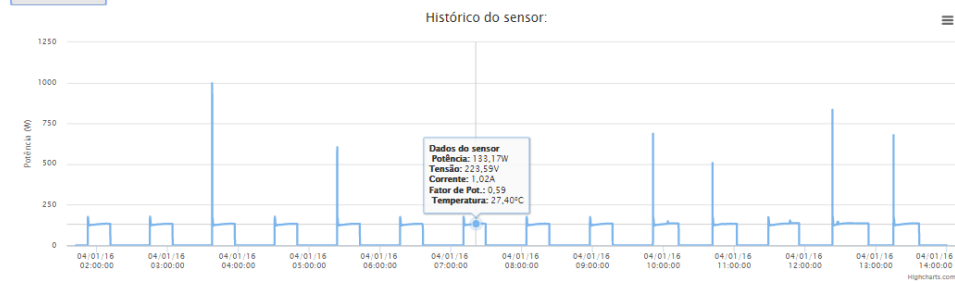
Essa página exibe os dados recebidos pelos dois XBEE NODES conectados a rede. As informações são atualizadas a cada cinco segundo.

[Mais informações »](#)

Visualizar historico do sensor

Selezione o sensor:
 A partir de:
 Até:

[Recuperar Dados](#)



Visualizar sensor ao vivo

Selezione o sensor
 Potência

[Visualizar](#)

Figura 30 - Página de controle mostrando dados de histórico

XBee Sensor Mesh Network

Essa página exibe os dados recebidos pelos dois *XBEE NODES* conectados a rede. As informações são atualizadas a cada cinco segundo.

[Mais informações »](#)

Sensores Conectados

Nome do sensor:

Status: **GRAVANDO**

Nome do sensor:

Status: **GRAVANDO**

© Company 2014

Figura 31 - Página de controle exibindo os Nodes conectados

5 COMISSONAMENTO DO PROTÓTIPO

Nesta fase, foi realizada a calibração e diversos testes para avaliar se o funcionamento do sistema está conforme o planejado.

Todos os sistemas são testados de forma integrada. São realizados, primeiramente, testes sem carga, e, se tudo estiver correto, os sistemas são comissionados como se estivessem em operação.

5.1 Calibração CS5490

O processo de calibração e compensação do CI CS5490 consiste de quatro processos, sendo três obrigatórios e um ocorrendo na dependência das condições medidas, este último, o de calibração de fase. Esta calibração só é necessária se a medição do fator de potência de uma carga aplicada puramente resistiva for diferente de 1. Durante a calibração dos sensores, ambos apresentaram valores iguais a 1, quando aplicada uma carga puramente resistiva, tornando-se assim, desnecessária a calibração de fase.

A Figura 32 mostra o *dataflow* de calibração.

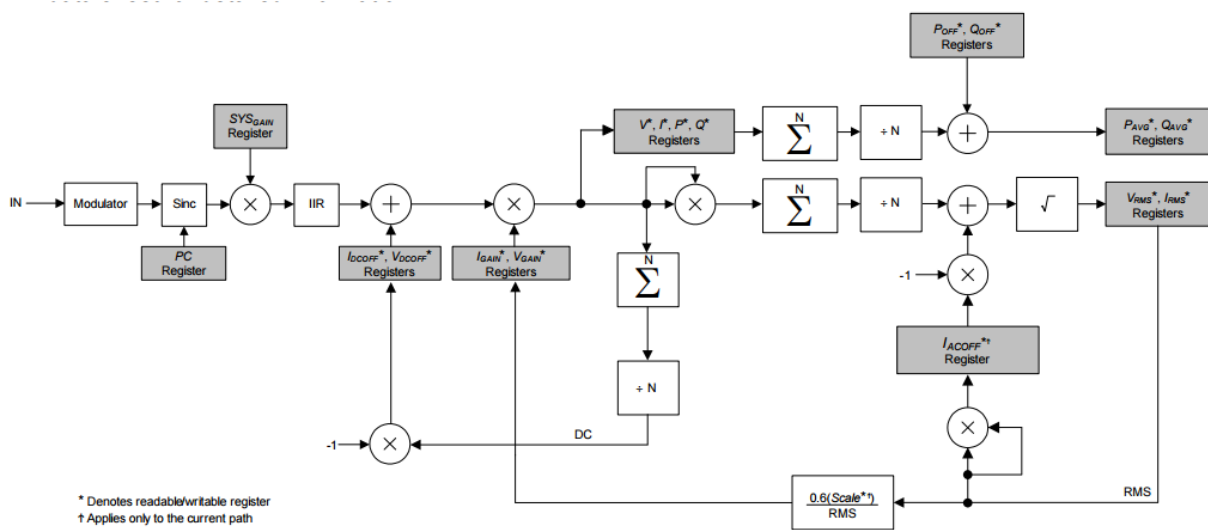


Figura 32 - *Dataflow* de Calibração (Cirrus Logic, 2012)

Existem sete registradores que podem ser utilizados na calibração do sistema: I_{DCOFF} , V_{DCOFF} , I_{GAIN} , V_{GAIN} , P_{OFF} , Q_{OFF} e I_{ACOFF} .

Os registradores I_{DCOFF} e V_{DCOFF} são utilizados para remover o componente DC da saída do ADC (do inglês *Analog-Digital Converter*), porém, em medidas de potência AC, os mesmos

são desnecessários, dando-se preferência ao filtro passa-alta que existe no circuito e pode ser ativado via registrador.

I_{GAIN} e V_{GAIN} são registradores de ganho utilizados para compensar variações de *hardware* e diferentes sensores utilizados pelo usuário, como o resistor *Shunt* ou o sensor Rogowski de corrente. A calibração de ganho deve ser efetuada de preferência com tensão e corrente em *full-scale*, porém o circuito oferece opções de calibração com valores menores que os máximos. Por limitações técnicas, a calibração foi feita com a tensão no valor máximo e corrente em aproximadamente 60% do valor máximo. A Figura 33 mostra o fluxograma completo de calibração.

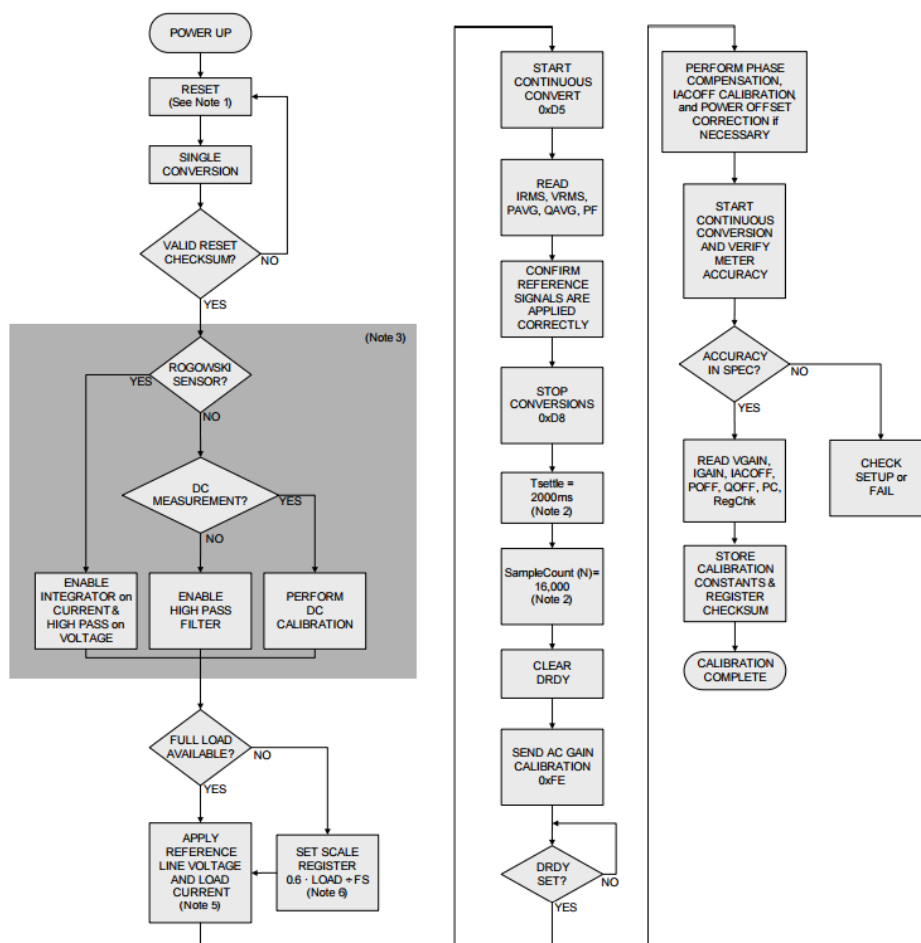


Figura 33 - Fluxograma completo de calibração (Cirrus Logic, 2012)

A calibração com tensão máxima e corrente média faz uso do registrador de escala da corrente (*Current scale register*), que é calculada pela equação abaixo:

$$I_{SCALE} = \frac{I_{REF}}{I_{MAX}} \cdot 0,6 \cdot 2^{23}$$

Equação 6 - Cálculo do valor do registrador de escala de corrente

Onde:

I_{SCALE} : Valor gravado no registrador de escala antes da calibração;

I_{MAX} : Valor máximo de corrente suportado pelo sensor;

I_{REF} : Valor de corrente utilizado na calibração.

Ao final da calibração de ganho, os registradores I_{GAIN} e V_{GAIN} possuem seus valores calibrados.

Após essa calibração ainda é possível que exista algum erro de *offset* no caminho AC do sensor. A calibração de *AC Offset* removerá esse desvio. Ao final da mesma, o registrador I_{ACOFF} guardará o valor calibrado. Vide fluxo mostrado na Figura 34.

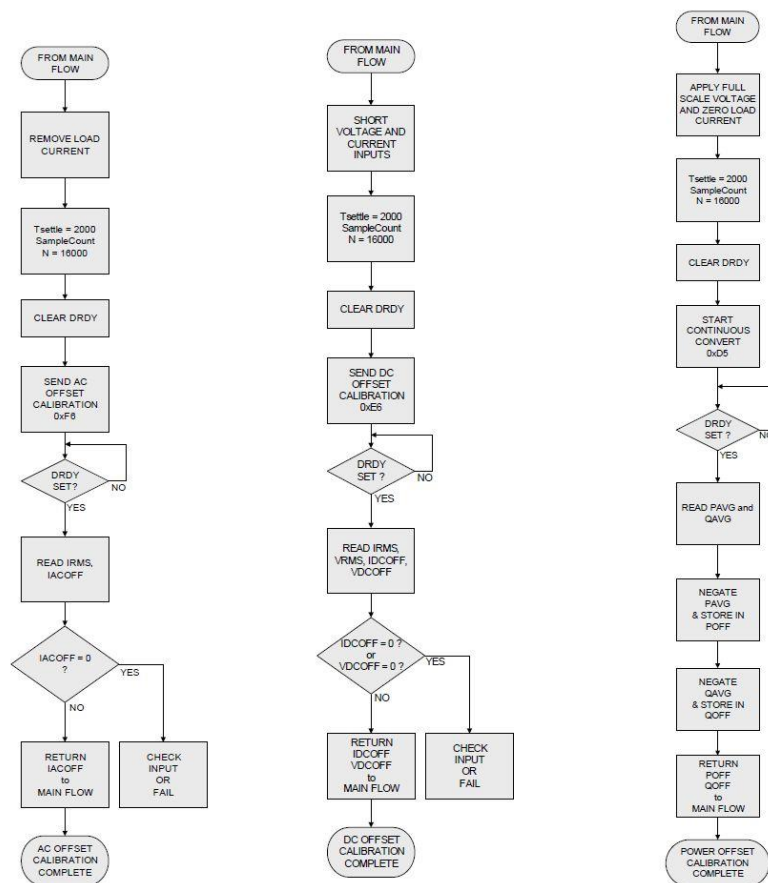


Figura 34 - Fluxos de calibração AC Offset, DC Offset e No Load Offset (Cirrus Logic, 2012)

A última calibração feita foi a *No Load Offset*. Os registradores que guardam os valores calculados de potência ativa média, P_{AVG} , e potência reativa média, Q_{AVG} , podem possuir desvios quando nenhuma carga é aplicada. A calibração sem carga remove esse desvio, e ao final os registradores P_{OFF} e Q_{OFF} possuem os valores dos desvios multiplicados por -1, que ao final do fluxo de cálculo é adicionado aos valores calculados. O fluxo de calibração pode ser visto na Figura 34.

A calibração dos sensores foi efetuada no laboratório de Instalações Elétricas e Eletricidade, situado no SG9 - Laboratório de Engenharia Mecânica da Universidade de Brasília. Lâmpadas incandescentes conectadas em série foram utilizadas como carga puramente resistiva, com potências variando entre 60W e 100W. A configuração pode ser vista na Figura 36. Para simular a tensão na escala total a mesma foi elevada de 220V para 250V utilizando um Varivolt (autotransformador monofásico), Figura 35. Nas medições, fez-se uso de dois multímetros, um conectado em paralelo na entrada do sensor para medir a tensão de entrada e outro conectado em série para medir a corrente que percorria o circuito. Os dados apresentados foram utilizados para todos os cálculos subsequentes. Com isso, a precisão total do sistema será considerada igual à precisão da escala utilizada no multímetro em questão.



Figura 35 – Varivolt - Autotransformador monofásico

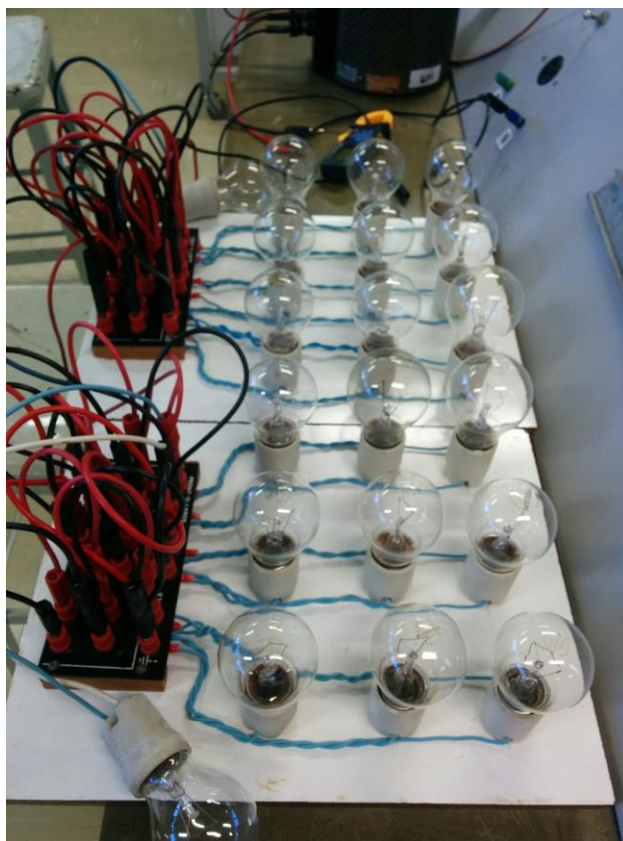


Figura 36 - Carga puramente resistiva

5.2 Testes

Testes de estabilidade foram realizados no conjunto como um todo. Tanto a Central como os Nodes mantiveram-se em funcionamento ininterrupto por duas semanas, sem nenhum problema encontrado.

Dos dois Nodes conectados, um foi usado para testes de estabilidade, conectado 100% do tempo, estando em funcionamento pelo mesmo período que a Central, sem apresentar nenhum problema. O segundo foi usado para testes de desconexão e reconexão, sendo desligado e ligado em intervalos aleatórios, com o sistema apresentando resultados esperados. Na desconexão, o sistema o retira do dicionário de Nodes conectados dentro do tempo esperado de 15 segundos. Na reconexão, o Node envia automaticamente uma mensagem ao coordenador, informando sua entrada na rede, e é imediatamente adicionado ao dicionário de Nodes. Todo esse processo de desconexão e reconexão, adição e retirada do dicionário acontece de forma transparente ao usuário e não influenciou de maneira perceptível os outros Nodes conectados na rede.

Também foram executados testes com redirecionamento de pacotes. Um dos Nodes foi afastado da central até que perdesse totalmente a conexão, em seguida o segundo Node foi

colocado entre a Central e o que está fora de alcance, e, em poucos segundos, a conexão foi reestabelecida com as mensagens voltando a ser entregues. Diversos testes foram realizados dessa forma durante as duas semanas em que o sistema ficou em funcionamento constante.

Testes de estabilidade na página de controle também foram realizados. Ela foi acessada e testada diversas vezes, em diferentes sistemas operacionais e navegadores, sempre se comportando da maneira esperada. Outro teste realizado foi a solicitação, por um período de 24 horas, de dados ao vivo, sem nenhum problema encontrado. Um máximo de cinco usuários conectados ao mesmo tempo foi testado, todos solicitando dados de histórico enquanto dados ao vivo de um Node eram mostrados. Nenhuma lentidão foi reportada, apesar de o consumo de recursos do microcomputador se ter elevado consideravelmente durante este teste.

Uma das preocupações nos testes foi quanto ao vazamento de memória em algum ponto do sistema, porém, como pode ser visto na Figura 37, após uma semana de funcionamento, o sistema não apresentava consumo de memória fora do esperado.

Testes de precisão também foram efetuados, onde os valores de tensão e corrente, calculados pelos Nodes, eram comparados aos valores disponibilizados por multímetros. Neste teste os valores apresentaram-se de acordo com o esperado. Porém, para que a calibração seja propriamente validada são necessários instrumentos de medição com precisão melhor do que os disponíveis durante o desenvolvimento do projeto.

Além das avaliações utilizando multímetros foram efetuados ensaios onde aparelhos presentes no dia-a-dia das pessoas foram utilizados para testar a precisão dos sensores. Todas as medições foram comparadas com os valores de consumo teórico, retirados diretamente dos aparelhos. Na Tabela 14 podem ser vistos os resultados medidos.

Tabela 14 - Potência nominal vs potência aferida

Aparelho	Potência nominal	Potência aferida
Torradeira ²³	850W	máx: 851.23W - min: 825.20W
Televisão LCD ²⁴	200W	máx: 210,74W - min: 129,85W ²⁵
Jarra Elétrica ²⁶	1850W	máx: 1814,5W - min: 1778.14W

De forma geral, o sistema como um todo mostrou-se estável e sem maiores problemas, recuperando-se após queda de energia e sempre respondendo da maneira esperada quando comandos eram gerados pela página de controle.

²³ Torradeira Philips Walita modelo RI2630

²⁴ Televisão Samsung LN40B550K1M

²⁵ A potência varia drasticamente com o volume do som e o nível de claridade da tela

²⁶ Jarra Elétrica Suggar modelo JR1702IX

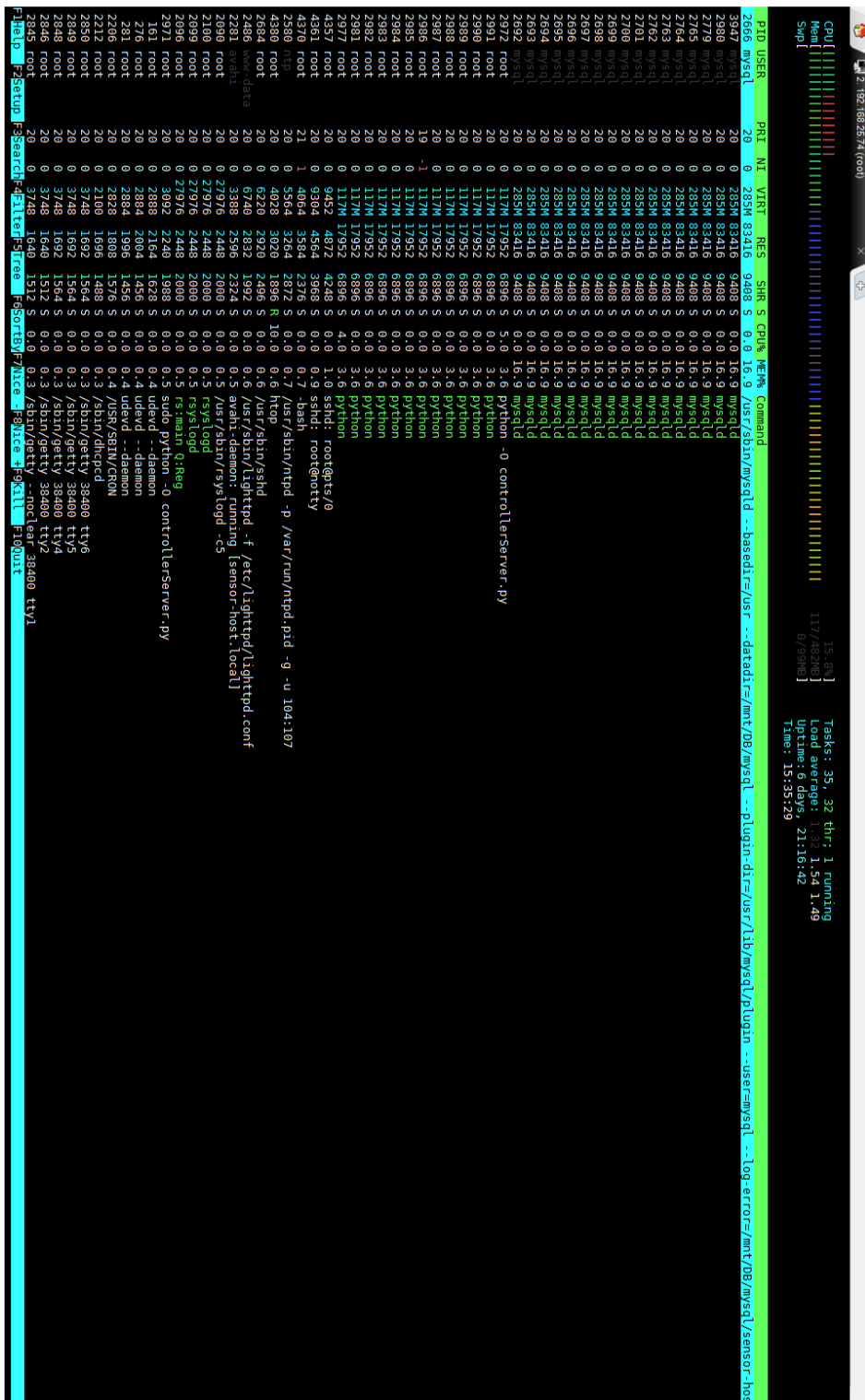


Figura 37 - Informações de consumo de recursos no Raspberry Pi®

6 CONCLUSÃO E RESULTADOS

O projeto e desenvolvimento do sistema proposto mostrou-se bastante laborioso. Diversos desafios foram encontrados no desenvolvimento, como a criação da placa de sensoriamento, que envolveu tensões muito acima daquelas já trabalhadas, soldagem de componentes SMD e o processo calibração.

O protótipo final correspondeu a todas as expectativas especificadas no início do projeto: robustez, precisão razoável, independência de internet para o correto funcionamento, redirecionamento de mensagens automático para utilização em grandes áreas, onde uma conexão WiFi comum não seria capaz de alcançar.

O preço final do sistema é compatível com o do mercado, estando abaixo ou no mesmo patamar de soluções comparáveis. Mesmo com um preço mais baixo o sistema aqui descrito tem ainda o potencial para superar os demais, visto que introduz diversas vantagens em relação aos sistemas atuais, como:

- Redirecionamento automático de mensagens;
- Alta capacidade de armazenamento de medições;
- Maior número de sensores conectados ao mesmo tempo;
- Conexão sem fio estável e robusta;
- Grande amplitude de valores de voltagem de entrada (90V – 250V);
- Alto valor de potência ativa máxima (3750W).

Uma segunda versão do módulo sensor pode ser desenvolvida para que o preço final do produto seja reduzido consideravelmente. Esta segunda versão do módulo sensor não utilizaria comunicação via protocolo ZigBee. Seria utilizada comunicação WiFi, possivelmente usando-se o módulo ESP8266, visto na Figura 38, um módulo que já possui isolamento eletromagnético, tem baixo custo, por volta de US\$3,00, e funciona em padrão 802.11n, com alcance teórico de 70 metros²⁷ em ambientes fechados, suficiente para a maioria dos locais onde seria utilizado.

²⁷ http://www.wireless-nets.com/resources/tutorials/migrate_80211n.html



Figura 38 - ESP8266MOD SMT (Shopclues, n.d.)

Com a internet cada vez mais presente no nosso dia a dia, a limitação da conexão constante é menos impeditiva.

Duas soluções seriam oferecidas: uma mais cara, onde não é necessário acesso à internet e outra mais barata, sem o uso da Central e com o módulo sensor com preço mais baixo, porém, necessitando conexão com a internet.

Essa necessidade de uma conexão constante com a internet pode ser remediada fazendo-se o uso de uma maior memória não volátil em cada um dos Nodes, sendo que, no momento em que a conexão fosse interrompida, o módulo sensor armazenaria as medições na memória e, ao ocorrer o restabelecimento da conexão, o sistema faria o *upload* dos dados armazenados para a nuvem.

6.1 Viabilidade e custos

O preço dos componentes mais importantes é mostrado na Tabela 15. Os demais itens são todos passivos e podem ser fabricados por diversos fornecedores. Será necessário um estudo mais cauteloso para que o melhor custo/benefício seja encontrado.

Tabela 15 - Preço dos componentes ativos²⁸

Módulo	Preço
ATMEGA328P-AU (Microcontrolador)	US\$ 1,84 @ 1000 unidades
CS5490-ISZ (Sensor de energia)	US\$ 1,85 @ 1000 unidades
LM2594H (Fonte chaveada)	US\$ 2,08 @ 1000 unidades
DS3231S#T&R (Relógio de tempo real)	US\$ 3,99 @ 1000 unidades
XB24-Z7WIT-004 (Módulo XBee®)	US\$ 17,00

²⁸ Cotação do dia 01/01/2016 pelo site <http://br.mouser.com/>

Raspberry Pi® (Microcomputador)	US\$ 35,00
---------------------------------	------------

O custo total do sistema composto de uma Central e cinco Nodes girando em torno de US\$ 200,00, já considerando um adicional para componentes passivos e placas de circuito.

Este valor pode ser reduzido adquirindo-se um volume maior de componentes e fazendo-se a compra daqueles mencionados na Tabela 15 de fornecedores chineses que possuem preço significativamente menor. Tal custo é bem atraente quando comparado a outras soluções já disponíveis no mercado, pois apresenta valor igual ou menor em configurações semelhantes, tendo diversas vantagens competitivas, como uma maior capacidade de armazenamento de dados, possibilidade de um maior número de Nodes conectados ao mesmo tempo, maior potência máxima de entrada, flexibilidade no controle e visualização dos dados e conexão sem fio robusta e estável, um ponto de reclamação em outros sistemas. Dois exemplos podem ser vistos abaixo.

- KILL-A-WATT® WIRELESS²⁹
 - Preço: US\$ 74,32 por um sensor e uma central, US\$ 23,99 por sensor extra
 - Conexão: rádio frequência
 - Armazenamento: sem informações claras
 - Acesso: apenas pela central física com display
- WeMo® Insight Switch³⁰
 - Preço: US\$ 68,92 por unidade
 - Conexão: WiFi
 - Armazenamento: não possui
 - Acesso: apenas por celular com aplicativo próprio

O sistema tem como característica principal a simplicidade de uso e a possibilidade de controlar e visualizar o consumo de uma única tomada, em comparação com o consumo total da residência, o que dificulta a identificação do consumo específico de um aparelho, informação importante para ser possível avaliar onde e quando os picos de consumo acontecem. Após a instalação, o usuário precisa apenas acessar seu navegador preferido para visualizar e controlar todos os Nodes conectados ao sistema.

²⁹ <http://www.smarthome.com/p3-international-p4200-kill-a-watt-wireless-energy-consumption-display-with-sensor.html>

³⁰ <http://www.smarthome.com/belkin-wemo-insight-switch.html>

O acesso pode ser feito a partir de qualquer dispositivo com conexão à rede local do usuário, sem a necessidade de aparelhos conectados diretamente ao computador. Existe também a possibilidade de acesso à rede interna por meio da internet, fazendo o uso de servidores DDNS^{31,32} onde o usuário tem a possibilidade de acessar o sistema de onde estiver, podendo, na versão final do produto, controlar os mais variados dispositivos, tais como condicionadores de ar, televisores, até mesmo preparar com antecedência seu banho em uma banheira.

³¹ https://pt.wikipedia.org/wiki/DNS_dinâmico

³² <http://www.noip.com/pt-BR>

7 REFERENCIAS BIBLIOGRAFICAS

Cirrus Logic. (2012). *AN366 - REV2*. Fonte: <http://www.cirrus.com/>:

<http://www.cirrus.com/en/pubs/appNote/AN366REV2.pdf>

Cirrus Logic. (2013). *CS5490 : Two Channel Energy Measurement IC*. Fonte: Cirrus Logic:

http://www.cirrus.com/en/pubs/proDatasheet/CS5490_F3.pdf

Cirrus Logic. (s.d.). *CRD5490-Z Power Monitor*. Fonte: Cirrus Logic:

http://www.cirrus.com/jp/pubs/rdDatasheet/CRD5490-Z_RD1.pdf

Craig, W. C. (s.d.). *ZigBee: "Wireless Control That Simply Works"*. Fonte: Zigbee Alliance:

<http://www.zigbee.org>

Digi International. (n.d.). *Digi International*. Retrieved from Digi International:

<http://www.digi.com/>

Digi International. (s.d.). *XBee® ZB - RF modules utilizing the ZigBee PRO Feature Set*.

Fonte: Digi International: http://ftp1.digi.com/support/documentation/90000976_T.pdf

EE Times. (s.d.). *ZigBee: Wireless Technology for Low-Power Sensor Networks*. Fonte:

<http://www.eetimes.com/>: <http://m.eet.com/media/1069127/zigbee3.gif>

Faludi, R. (2011). *Building Wireless Sensor Networks: with ZigBee, XBee, Arduino, and Processing*. O'Reilly.

Farahani, S. (2008). *ZigBee Wireless Networks and Transceivers*. Newnes.

Fortytwoandnow. (s.d.). *Stellaris Launchpad - NRF24L01 radio - Part 1*. Fonte:

<http://fortytwoandnow.blogspot.com.br/>: <http://2.bp.blogspot.com/->

bVEADNjx5b4/U1tfuOYi7AI/AAAAAAAAAoI/CP5tlUy0TKE/s1600/multiceiver.png

Gravitech Us. (s.d.). *XBee ZB ZigBee Mesh Module 2.4GHz 2mW with Wire Antenna*. Fonte:

<http://www.gravitech.us/>: http://site.gravitech.us/MicroResearch/Wireless/XBee-MESH-W/XBee-MESH-W_1R.jpg

Home Coder. (s.d.). *Getting to grips with a Real Time Clock*. Fonte:

<https://homecoder.wordpress.com/>:

<https://homecoder.files.wordpress.com/2013/09/ds3231sn.jpg>

Home Toys. (s.d.). *IEEE 802.15.4 AND ZIGBEE COMPLIANT RADIO TRANSCEIVER*

DESIGN. Fonte: <http://www.hometoys.com/>:

http://www.hometoys.com/htinews/feb05/articles/chipcon/zigbee_files/image002.gif

KAUR, G., & AHUJA, K. (2011). *QoS measurement of Zigbee home automation network using various modulation schemes*. International Journal of Engineering Science and Technology (IJEST).

LABIOD, H., AFIFI, H., & DE SANTIS, C. (2007). *Wi-fi, Bluetooth, ZigBee and WiMax*.

Paris: Springer.

Mikrocontroller Praxis. (s.d.). *Serial Port BLE Module*. Fonte: [http://mikrocontroller-](http://mikrocontroller-praxis.de)

[praxis.de](http://mikrocontroller-praxis.de): <http://mikrocontroller-praxis.de/media/images/info/hm-10.jpg>

Million Bitz. (s.d.). *XBee API Frame Data transmit*. Fonte: <http://www.millionbitz.com/>:

<http://lh3.ggpht.com/->

[QkSBleZQ1CA/UxvXvsmZF3I/AAAAAAAAAQY/Zc3jt2FlSPM/image_thumb%25255B6%25255D.png](http://lh3.ggpht.com/-QkSBleZQ1CA/UxvXvsmZF3I/AAAAAAAAAQY/Zc3jt2FlSPM/image_thumb%25255B6%25255D.png)

Piyare, R., & Lee, S.-r. (Abril de 2013). *Performance Analysis of XBee ZB Module Based*.

Fonte: International Journal of Scientific and Engineering Research:

<http://www.ijser.org/researchpaper/Performance-Analysis-of-XBee-ZB-Module-Based-Wireless-Sensor-Networks.pdf>

RASPBERRY PI FOUNDATION. (s.d.). *NEW PRODUCT LAUNCH! INTRODUCING*

RASPBERRY PI MODEL B+. Fonte: <https://www.raspberrypi.org/>:

https://www.raspberrypi.org/wp-content/uploads/2014/07/rsz_b-.jpg

Rinky-Dink Electronics. (s.d.). *DS3231*. Fonte: <http://www.rinkydinkelectronics.com/>:

<http://www.rinkydinkelectronics.com/images/libpics/L0073P001408293553.png>

Shopclues. (s.d.). *ESP8266 ESP-12 WIFI Serial Wireless Transceiver Module*. Fonte:

<http://www.shopclues.com/>:

http://cdn.shopclues.net/images/detailed/25123/ESP12SunRobotics_1443028670.jpg

Sparkfun Electronics. (s.d.). *PCDUINO LITE - DEV BOARD*. Fonte:

<https://www.sparkfun.com>: <https://cdn.sparkfun.com/assets/parts/8/6/9/3/12077-03.jpg>

Sparkfun Eletronics. (s.d.). <https://www.sparkfun.com/products/12820>. Fonte: Sparkfun:

<https://cdn.sparkfun.com/assets/parts/8/6/8/0/12820-01.jpg>

Teixeira, L. M. (2006). Desenvolvimento de uma Aplicação com o Protocolo ZigBee aplicado em Instrumentação de Ensaio de Vão. São José dos Campos.

Texas Instruments. (2013). *LM2594/LM2594HV SIMPLE SWITCHER® Power Converter 150 kHz 0.5A Step-Down*. Fonte: Texas Instruments:

<http://www.mouser.com/ds/2/405/lm2594hv-405259.pdf>

THE INTENTIONAL GEEK. (s.d.). *Building a Cross-Compilation Environment for the BeagleBone*. Fonte: <http://www.andrewjarrell.com/>:

https://arnisandy.files.wordpress.com/2013/07/beaglebonetop_lrg.jpg

Titus, J. A. (2012). *The Hands-on XBEE Lab Manual: Experiments that Teach you XBEE Wireless Communications*. Newnes.